이학 박사 학위논문

Mathematical Analysis of Cryptographic Multilinear Maps

(암호학적 다중선형함수의 수학적 분석에 관한 연구)

2017년 8월

서울대학교 대학원

수리과학부

이창민

Mathematical Analysis of Cryptographic Multilinear Maps

(암호학적 다중선형함수의 수학적 분석에 관한 연구)

지도교수 천정희

이 논문을 이학 박사 학위논문으로 제출함

2017년 4월

서울대학교 대학원

수리과학부

이창민

이창민의 이학 박사 학위논문을 인준함

2017년 6월

위 원 장	김 명 환	(인)
부 위 원 장	천 정 희	(인)
위 원	David Donghoon Hyeon	(인)
위 원	권 순 학	(인)
위 원	윤 아 람	(인)

Mathematical Analysis of Cryptographic Multilinear Maps

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
to the faculty of the Graduate School of
Seoul National University

by

Changmin Lee

Dissertation Director : Professor Jung Hee Cheon

Department of Mathematical Sciences
Seoul National University

August 2017

Abstract

Mathematical Analysis of Cryptographic Multilinear Maps

Changmin Lee
Department of Mathematical Sciences
The Graduate School
Seoul National University

Multilinear maps are a very powerful tool in cryptography. Nonetheless, to date, only three types of multilinear maps have been published relying on a graded encoding scheme. The first candidate is proposed by Garg, Gentry, and Halevi (GGH) relying on an ideal lattice [GGH13a], the second one is defined on integers as established by Coron, Lepoint, and Tibouchi (CLT) [CLT13], and the last one is provided by Gentry, Gorbunov, and Halevi (GGH15) relying on a graph induced graded encoding scheme [GGH15]. These multilinear maps have led to a number of applications in cryptography such as one round key exchange protocol, witness encryptions, and even indistinguishable obfuscations. The security of the applications depends on some hardness problems derived from a graded encoding scheme.

However, none of them have reduction to well-known hard problems. For that reasons, many researches attempt to investigate the hardness of the problems. Actually, when low-level encodings of zero are given, the GGH scheme is known to be insecure by Hu and Jia [HJ16] and the last candidate of a multilinear map GGH15 is known to be insecure [CLLT16].

In the thesis, we describe an algebraic analysis on the hardness problems of two GGH and CLT multilinear maps. Common to two candidates are

constructed by graded encoding schemes and provide an additional public information zerotesting parameter, which is used to determine whether the hidden message is zero or not. Exploiting the structure of graded encoding scheme and additional input, we study how to solve the hardness problems in three cases.

First, we show another approach to break the GGH scheme with low level encodings of zero. According to the original GGH paper, finding a short vector for a given principal ideal lattice enables to break the scheme. Therefore, the parameters are set to be invulnerable to the best known algorithm for finding a short vector on ideal lattice. By proposing an improved lattice reduction algorithm to find a short vector, we prove that the multilinear map is broken within quasi polynomial time of the suggested parameters.

Second, we describe that how to construct a level-0 encoding of zero from GGH public parameter without level encodings of zero in the quasi-polynomial time of the suggested parameters. The obtained encoding of zero serves as a low level encoding of zero in the first study. Thus we also show that GGH without low level encodings of zero is insecure.

Finally, for CLT scheme with low level encodings of zero, we attempt to reveal the all secret elements of scheme in polynomial time. By multiplying encodings of zero to zerotesting parameter appropriately, one can obtain an integer matrix of secret quantities. Next we recover the secret elements by computing eigenvalues.

Key words: GGH multilinear maps, SVP, NTRU, ideal lattice, CLT multilinear maps, CRT-ACD
Student Number: 2012-20254

Contents

Chapter 1

Introduction

1.1 Multilinear maps

Multilinear maps are one of the important primitives in cryptography. Since Boneh and Silverberg formalized cryptographic multilinear maps in 2003 [BS03], various cryptographic applications have been proposed using multilinear map such as one round multi-party key exchange protocol, a witness encryption scheme, and indistinguishable obfuscation [BS03, GLW14, GGH$^+$13b]. Because these applications are a long standing open problems, designing multilinear maps has become an important research topic. Unfortunately, for a long time, it fails to produce secure multilinear maps and such applications could not actually be realized.

A decade later, in 2013, Garg, Gentry Halevi [GGH13a] introduced the new concept of (approximate) graded encoding scheme and proposed the construction of multilinear maps by using the graded encoding scheme. Moreover, by designing the graded encoding schemes relying on an ideal lattice they described the first candidate of multilinear maps as GGH scheme. Soon after, Coron *et al.* introduced another candidate of multilinear map by designing a new graded encoding schemes over the integers as CLT scheme [CLT13]. Both schemes are share similarity, which one is to provide extra public data called the zero-testing or extraction parameter, which allow to publicly de-

cide whether the message of the given encoding is zero or not. The GGH scheme and CLT scheme lead us to realize the various applications based on multilinear maps.

The security of the applications rely on 4 types of new hardness assumptions such as Graded Decisional Diffie-Hellman assumption (GDDH), Decisional Linear problem (DLIN), Subgroup Membership problem (SubM), and Graded External Decision Diffie-Hellman problem (GXDH), which are served from graded encoding schemes. These problems have been initially used in the context of cryptographic bilinear maps [Sco02, BBS04, BGN05].

Unfortunately, in case of the GGH scheme with low grade encoding of zeros (i.e. low level encoding of zeros), it was realized that the hardness problems could be broken in polynomial-time by using called zeroizing attack. [HJ16, GGH13a]. Briefly, The attack consists in multiplying the encoding of zero and zero-testing parameters for a given encoding of $\mathbf{m}$ allows us to obtain multiples of $\mathbf{m}$ exactly. This value serves to solve all related problems of GGH schemes. The more details were described in original paper [HJ16, GGH13a].

On the other hand, in the case of CLT, since the attack algorithm using the encoding of zero is not known, the presumed hardness of the CLT instantiations of GDDH, SubM, DLIN and GXDH was exploited as a security grounding for several cryptographic constructions [ABP15, BLMR13, GLSW15, Zim15]. Whereas the both schemes without low grade encodings of zero are still known to be hard.

Soon after, a third candidate construction of a variant of graded encoding schemes was proposed in [GGH15]. And at a similar time, a new CLT multilinear maps [CLT15] are proposed. Unfortunately, these schemes are also known to be insecure [CLLT15, CFL$^+$16].

1.2 Contributions

In this thesis, we provide an algebraic analysis of the hardness problems depended on multilinear maps. First of all, we abstract the problems of the GGH scheme to shortest vector problem on ideal lattice or Overstretched NTRU problem. In case of CLT scheme, we reduce the CRT-ACD with auxiliary input. Next we describe an analysis of the problems.

1.2.1 Analysis of the GGH scheme

GGH with low level encodings of zero

As mentioned above, the GGH scheme with low level encodings of zero was broken by Hu and Jia [HJ16]. In this thesis, we provide another approach to analysis of the scheme. According to original GGH paper [GGH13a, Sec. 6.3.3], for a secret element $\mathbf{g}$ of the GGH scheme, if one can finds a short element of $\mathbf{b} \in \langle \mathbf{g} \rangle$ such that $\|\mathbf{b}\| \leqslant q^{1/4}$, the GDDH problems is not secure. In other words, the GDDH problem are reduced to shortest vector problem (SVP) on lattice. More precisely, let R be a ring $\mathbb{Z}[X]/\langle X^n + 1 \rangle$ and $\langle \mathbf{g} \rangle$ an ideal generated by $\mathbf{g} = \sum_{i=0}^{n-1} g_i \in R$. For a generator $\mathbf{g}$, by identifying it to a vector $(g_0, g_1, \cdots, g_{n-1})$, the size of $\mathbf{g}$, $\|\mathbf{g}\|$, is defined by $\sqrt{\sum_{i=0}^{n-1} g_i^2}$. Moreover, one can consider the principal ideal $\langle \mathbf{g} \rangle$ as a lattice generated by $\{\mathbf{g}, \mathbf{g} \cdot X, \cdots, \mathbf{g} \cdot X^{n-1}\}$. Given a basis of the lattice $\langle \mathbf{g} \rangle$, SVP on ideal lattice is to find an short element $\mathbf{b}$ of an ideal $\langle \mathbf{g} \rangle$.

As the first contribution of this thesis, we improve a lattice reduction algorithm to find a short vector. The algorithm is called by sublattice algorithm. Generally, a lattice basis reduction algorithm $\mathcal{A}_\delta$ with a root Hermite factor δ on n-dimensional lattice $\mathcal{L}$ outputs a lattice point $\mathbf{b} \in \mathcal{L}$ such that $\|\mathbf{b}\| \leqslant \delta^n \cdot \det(\mathcal{L})^{1/n}$. When the determinant of an integral lattice is less than δ^{n^2}, we propose an algorithm to obtain a lattice point $\mathbf{b}$ such that $\|\mathbf{b}\| \leqslant 2^{2 \cdot \sqrt{\log \delta \cdot \log \det \mathcal{L}}}$. This algorithms run in time of algorithm $\mathcal{A}_\delta$ with poly-

nomial time of the dimension n and $\log(\det \mathcal{L})$. The main idea is to reduce a SVP on a lattice L to one on its sublattice whose determinant is bounded by that of L. This technique was used to orthogonal lattice attacks on Learning with Errors or Small Integer Solution problems where the lattice dimension can be taken flexible without increasing determinant [MR09] Our idea is not bounded to specific attacks and can be applied to any integral lattices.

In general, the determinant of a sublattice may be smaller or larger than that of the original lattice. Our idea is to use the Hermite normal form (HNF) to obtain a sublattice with bounded determinant. The HNF of an integral lattice is a unique (generalized) lower-triangular matrix, the columns of which form a basis of the lattice and it can be computed in polynomial time from any basis of the lattice [MW01]. We show that the sublattice generated by the last m columns has a smaller determinant than that of the original lattice for any positive integer $m \leqslant n$. For example, by applying LLL algorithm to this sublattice of appropriate m, we obtain a short vector, the size of which is upper-bounded by $2^{\sqrt{\log \det L}}$, which is smaller than the shortest vector from LLL when the determinant of L is smaller than $2^{n^2/4}$. We may apply any approximate SVP algorithm to our algorithm instead of LLL, including the BKZ algorithm. Our algorithm with BKZ of block size $\mathsf{polylog}(\lambda)$ breaks the GGH scheme with the security parameter λ.

GGH with top level encodings of zero

In case of the GGH scheme with top level encodings of zero, we observe that the GDDH problem is transformed to Overstretched NTRU problem (ONTRU). Let $\mathbf{f}$ and $\mathbf{g}$ be polynomials of a bounded Euclidean coefficient in the ring $\mathbb{Z}[X]/\langle X^n + 1\rangle$ with a power of two n. Given the polynomial $[\mathbf{f}/\mathbf{g}]_q \in \mathbb{Z}_q[X]/\langle X^n + 1\rangle$, the NTRU problem is to find $\mathbf{a},\ \mathbf{b} \in \mathbb{Z}[X]/\langle X^n + 1\rangle$ with a small Euclidean coefficient such that $[\mathbf{a}/\mathbf{b}]_q = [\mathbf{f}/\mathbf{g}]_q$. When, q is a super polynomial on security parameter λ, it is called by ONTRU problem and which is introduced in [ABD16].

In order to solve the ONTRU, we provide so called subfield algorithm and

hybrid algorithm combining the subfield and sublattice algorithm. In detail, when n is $O(\log^2 q)$ and $\|\mathbf{g}\|, \|\mathbf{f}\|, \|\mathbf{g}^{-1}\|$ are $O(n)$, we propose an algorithm to solve the ONTRU problem, which runs in $2^{O(\log^2 q)}$ time. The main technique of our algorithm is the reduction of a problem on a field to one in a subfield with trace map. Concurrently and independently, Albrecht *et al.* suggest a very similar algorithm and hold same results by using norm map instead of trace [ABD16]. In the GGH scheme without low-level encodings of zero, our algorithm can be directly applied to attack this scheme if we have some top-level encodings of zero. Using our algorithm, we can construct a level-0 encoding of zero and utilize it to attack a security ground of this scheme in the quasi-polynomial time of its security parameter suggested by [GGH13a]

In addition, we suggest an improvement of the subfield attack by combining the sublattice algorithm and subfield algorithm. Our attack reduces the required block sizes of the BKZ algorithm from β to β' satisfying $\frac{\beta'}{\log \beta'} \sim \frac{27n \log M}{2 \log^2 q}$, while $\frac{\beta}{\log \beta} \sim \frac{16n \log M}{\log^2 q}$ in the previous. Our Hybrid attack implies that the degree should be increased from n to $\frac{32n}{27}$ in order to maintain the same security level in the cryptosystems based on the ONTRU problems.

1.2.2 Analysis of the CLT scheme

Although the CLT scheme also exposes low level encodings of zero, the scheme has no way of attacking with zero encodings. In this thesis, we first define a new problem CRT-ACD with auxiliary input (CRT-ACDwAI) and reduce from the CLT scheme with low level encodings of zero to CRT-ACDwAI.

The CRT-ACDwAI is to find η-bit primes p_i for all $1 \leqslant i \leqslant n$ for given many samples in the form of $\mathsf{CRT}_{(p_1,\cdots,p_n)}(r_1, \cdots, r_n)$ which is an integer congruent to integer r_i in modulo p_i, $x_0 = \prod_{i=1}^{n} p_i$ and $\hat{P} = \sum_{i=1}^{n} x_0/p_i$.

When $3 \log |r_i| + 1 < \eta$, we describe an algorithm to solve the CRT-ACDwAI. The algorithm runs in polynomial-time and allows one to publicly compute all p_i, which is also secret parameters of the CLT scheme. By adapting the zeroizing attack to CLT scheme, one can reduce from public parameters of CLT scheme to CRT-ACDwAI. It implies that any quantities involving

the kept secret of the CLT scheme are recovered. Additionally we describe another algorithm. It runs also in polynomial-time and reveals the $\prod_{i=1}^{n} r_i$ and can be used to solve the SubM, DLIN, GXDH for CLT scheme.

List of Papers. This thesis contains results of sublattice algorithm originally appeared in [CL15]. The attack algorithm for NTRU problem, subfield algorithm, that were obtained jointly with Jung Hee Cheon and Jin Heok Jung [CJL16], which was presented in Algorithmic Number Theory Symposium (ANTS 2016). It also contains the result of the improved subfield algorithm by [CHL17].

In addition, this thesis includes a joint work with Jung Hee Cheon, Hansol Ryu, Kyoohyung Han, and Damien Stehle [CHL$^+$15], which published in Eurocrypt 2015 and its variant version.

- [CJL16] Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low level encoding of zero. *Mh*, 1:0, 2016.
- [CL15] Jung Hee Cheon, and Changmin Lee. Approximate algorithms on lattices with small determinant. IACR Cryptology ePrint Archive, Report 2015/461, 2015.
- [CHL17] Jung Hee Cheon, Minki Hhan, and Changmin Lee: Cryptanalysis of the Overstretched NTRU Problem for General Modulus Polynomial, preprint.
- [CHLRS15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé: Cryptanalysis of the multilinear map over the integers. In *Advances in Cryptology - EUROCRYPT 2015*, pages 3–12, 2015.
- [CHLRS15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé: Cryptanalysis on the Multilinear Map over the Integers and its Related Problems, preprint.

Chapter 2

Preliminaries

First of all, we define some notations to be used and basic notions related to multilinear maps.

2.1 Notations.

For an integer q, we use the notation $\mathbb{Z}_q := \mathbb{Z}/(q\mathbb{Z})$ and $[R]_q := \mathbb{Z}_q[X]/\langle X^n + 1\rangle = R/qR$. We denote the number in $\mathbb{Z}_q$ within the range $\left(-\frac{q}{2}, \frac{q}{2}\right]$ by $(x \bmod q)$ or $[x]_q$, which is congruent to x modulo q. For $\mathbf{u} = \sum_{i=0}^{n-1} u_i X^i \in R$, $[\mathbf{u}]_q = \sum_{i=0}^{n-1} [u_i]_q X^i$ and $\|\mathbf{u}\|$ denote the Euclidean norm of $\mathbf{u}$.

We define $\iota : \mathbb{Z}_q \longrightarrow \mathbb{Z}$ by $[x]_q \in \mathbb{Z}_q \mapsto x \in \mathbb{Z}$ for $-\frac{q}{2} < x \leqslant \frac{q}{2}$. We extend this map to $[R]_q$ by applying it to each coefficient. By abuse of notation, we omit ι unless it will be confused when identifying $[x]_q \in \mathbb{Z}_q$ with an integer x when $-\frac{q}{2} < x \leqslant \frac{q}{2}$.

Throughout this paper, we assume that an integer n is a power of 2. Then, $K := \mathbb{Q}[X]/\langle X^n + 1\rangle$ is a number field with the ring of integers $R := \mathbb{Z}[X]/\langle X^n + 1\rangle$. In particular, K is a Galois extension of $\mathbb{Q}$, and we denote the Galois group of K over $\mathbb{Q}$ by $\mathrm{Gal}(K/\mathbb{Q})$. As in the technical overview, for any polynomial $\mathbf{h} = \sum_{i=0}^{n-1} h_i X^i \in K$, we consider it to be a column vector $(h_0, \cdots, h_{n-1})^T$. When we need an inverse of an element $\mathbf{a} \in R$, we

usually consider the inverse in K with the notation $\mathbf{a}^{-1}$. If we want to consider it in $[R]_q$ and not in K, then we denote it by $[\mathbf{a}^{-1}]_q$. We use bold letters to denote vectors or ring elements in $\mathbb{Z}^n$ or R.

2.2 Graded encoding Schemes and Multilinear map Procedure.

We first recall the formal definition of multilinear map suggested by Boneh and Silverberg [BS03].

Definition 2.2.1 (Multilinear maps [BS03])**.** For κ+1 cyclic groups $G_1, \cdots, G_\kappa$, and G_T of the same order p, an κ-multilinear map $e : G_1 \times \cdots \times G_\kappa \to G_T$ has the following properties:

1. For an element $a \in \mathbb{Z}_p$ and $\{g_i \in G_i\}_{1 \leqslant i \leqslant \kappa}$, it holds:

$$e(g_1, \cdots, a \cdot g_i, \cdots, g_\kappa) = a \cdot e(g_1, \cdots, g_i, \cdots, g_\kappa).$$

2. If the elements g_i is a generator of G_i, respectively, then $e(g_1, \cdots, g_\kappa)$ is a generator of G_T.

In the original definition [BS03], the cyclic group are set as same group $G_1 = G_2 = \cdots = G_\kappa$, which is called symmetric multilinear maps. The asymmetric multilinear maps with different G_i's are considered in [GGH13a]. Throughout this thesis, we generally deal with symmetric case, and only with asymmetric case if necessary.

Unfortunately, when $\kappa \geqslant 3$, none of the multilinear maps are published matched to the formal definition of multilinear maps. At 2013, instead of a formal definition, Garg *et al.* proposed a new concept "graded encoding scheme" and proposed an approximate multilinear maps by exploiting the graded encoding scheme. After that, GGH and CLT (approximate) multilinear maps are provided by constructing a graded encoding scheme relying on

ideal lattice and integers, respectively. Now we describe the formal definition of a κ-graded encoding scheme referring to original paper [GGH13a].

Definition 2.2.2 (Graded Encoding Scheme [GGH13a])**.** A κ-Graded Encoding Scheme for a commutative ring R is a system of sets $S = \{S_i^{(\alpha)} \in \{0,1\}^* : 1 \leqslant i \leqslant \kappa, \alpha \in R\}$, with the following properties

1. For every $1 \leqslant i \leqslant \kappa$, the sets $\{S_i^{(\alpha)} : \alpha \in R\}$ are disjoint.

2. There are binary operations $+$ and $-$ (on $\{0,1\}^*$) such that for every $\alpha_1, \alpha_2 \in R$, every $1 \leqslant i \leqslant \kappa$, and every $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_i^{(\alpha_2)}$, it holds that

$$u_1 + u_2 \in S_i^{(\alpha_1+\alpha_2)} \text{ and } u_1 - u_2 \in S_i^{(\alpha_1-\alpha_2)},$$

   where $\alpha_1 + \alpha_2$ and $\alpha_1 - \alpha_2$ are addition and subtraction in R.

3. There in an associative binary operation $\times$ (on $\{0,1\}^*$) such that for every $\alpha_1, \alpha_2 \in R$, every $1 \leqslant i, j \leqslant \kappa$ with $0 \leqslant i + j \leqslant \kappa$, and every $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_j^{(\alpha_2)}$, it holds that

$$u_1 \times u_2 \in S_{i+j}^{(\alpha_1 \cdot \alpha_2)},$$

   where $\alpha_1 \cdot \alpha_2$ is multiplication in R.

Throughout this thesis, we denote a level-i encoding of $\mathbf{m}$ as $\mathsf{enc}_t(\mathbf{m})$. The main difference between multilinear maps and graded encoding scheme is two fold; one is that the encodings has a level (or grade). The set S_i^{α} means that level i encoding of α. The second main difference is that the encodings in graded encoding scheme are randomized whereas the encodings of multilinear maps are deterministic. It implies that a ring element α can be encoded many ways.

Since the image of multilinear map is a deterministic function for a ring element α's only, it needs additional procedure to erase the randomness.

In [GGH13a] and [CLT13], by providing a zerotesting procedure, they delete the randomness and enable to construct a multilinear map. Now we describe a multilinear map procedure using graded encoding scheme; we refer to [GGH13a] As previously we consider only the symmetric case.

Multilinear map Procedure.

Instance Generation. The randomized $\mathsf{InstGen}(1^\lambda, 1^\kappa)$ takes as inputs the parameters λ and κ, and outputs (params, $\mathbf{p_{zt}}$), where params is a description of a κ-Graded Encoding System as above, and pzt is a zero-test parameter.
Ring Sampler. The randomized $\mathsf{samp(params)}$ outputs a level-zero encoding $a \in S_0^\alpha$ for a nearly uniform element $\alpha \in_R R$. Note that the encoding a does not need to be uniform in S_0^α.
Encoding. The (possibly randomized) $\mathsf{enc(params}, a)$ takes as input a level-zero encoding a S_0^α for some $\alpha \in S_0^\alpha$, and outputs the level-one encoding $u \in S_1^\alpha$ for some α.
Re-Randomization. The randomized $\mathsf{reRand(params}, i, u)$ re-randomizes encodings relative to the same level i. Specifically, given a level-l encoding $u \in S_l^{(\alpha)}$, it outputs another encoding $u_0 \in S_l^{(\alpha)}$. Moreover for any two $u_1, u_2 \in S_l^{(\alpha)}$, the output distributions of $\mathsf{reRand(params}, i, u_1)$ and $\mathsf{reRand(params}, i, u_2)$ are nearly the same.
Addition and negation. Given params and two encodings relative to the same level, $u_1 \in S_l^{(\alpha_1)}$ and $u_2 \in S_l^{(\alpha_2)}$, we have $\mathsf{add(params}, u_1, u_2) \in S_l^{(\alpha_1+\alpha_2)}$ and $\mathsf{neg(params}, u_1) \in S_l^{(-\alpha_1)}$. Below we write $u_1 + u_2$ and $-u_1$ as a shorthand for applying these procedures.
Multiplication. For $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_j^{(\alpha_2)}$, we have $\mathsf{mul(params}, u_1, u_2) = u_1 \times u_2 \in S_{i+j}^{(\alpha_1 \cdot \alpha_2)}$.
Zero-test. The procedure $\mathsf{isZero(params}, \mathbf{p_{zt}}, u)$ outputs 1 if $u \in S_\kappa^{(0)}$ and 0 otherwise.
Extraction. The procedure extracts a random function of ring elements from their level-κ encoding. Namely $\mathsf{ext(params}, \mathbf{p_{zt}}, u)$ outputs $s \in \{0,1\}^\lambda$, such that:

1. For any $\alpha \in R$ and $u_1, u_2 \in S_\kappa^{(\alpha)}$, $\mathsf{ext}(\mathsf{params}, \mathbf{p_{zt}}, u_1) = \mathsf{ext}(\mathsf{params}, \mathbf{p_{zt}}, u_2)$.

2. The distribution $\{\mathsf{ext}(\mathsf{params}, \mathbf{p_{zt}}, u) : \alpha \in_R R, u \in S_\kappa^{(\alpha)}\}$ is nearly uniform over $\{0,1\}^\lambda$.

In the original [GGH13a] paper, the authors provide a slightly weak definitions of isZero and ext, where isZero can output 1 even for top level encoding of some nonzero element with negligible probability, and ext can output different outputs for top level encodings of same element, also with negligible probability. In this sense, a multilinear map relying on a graded encoding scheme is a approximate function.

2.3 Hardness Problems.

Finally, we recall the hardness problems and its related problems for multilinear maps underlying on graded encoding scheme. described in [GGH13a]. First, we introduce a hardness assumption for multilinear maps, graded decisional Diffie-Hellman (GDDH) problem. Briefly, given a set of $\kappa+1$ level-one encodings of random elements, it is to distinguish between a level-κ encodings of their product and a random element.

Definition 2.3.1 (Graded Decisional Diffie-Hellman (GDDH))**.** For an adversary $\mathcal{A}$ and parameters λ, κ we consider the following process:
Generating public parameter: $(\mathsf{params}, \mathbf{p_{zt}}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^\kappa)$
Sampling: Choose $a_j \leftarrow \mathsf{samp}(\mathsf{params})$ for all $0 \leqslant j \leqslant \kappa$
Encoding: Set $u_j \leftarrow \mathsf{reRand}(\mathsf{params}, 1, \mathsf{enc}(\mathsf{params}, 1, a_j))$ for all $0 \leqslant j \leqslant \kappa$
Random sampling: Choose $b \leftarrow \mathsf{samp}(\mathsf{params})$
Right product at level-κ: Set $u = \mathsf{reRand}(\mathsf{params}, \kappa, \prod_{j=0}^{\kappa} a_j)$
Random value at level-κ: Set $\hat{u} = \mathsf{reRand}(\mathsf{params}, \kappa, b)$

The GDDH problem is to distinguish between two distributions, $\mathcal{D}_{DDH}$ and

$\mathcal{D}_R$, where

$$\mathcal{D}_{DDH} = \{\mathsf{params}, \mathbf{p}_{zt}, u_0, \ldots, u_\kappa, u\} \text{ and } \mathcal{D}_R = \{\mathsf{params}, \mathbf{p}_{zt}, u_0, \ldots, u_\kappa, \hat{u}\}.$$

Remark As mentioned previously, the many applications of multilinear maps rely on several problems such as SubM, DLIN, GXDH. In the first public version of [GGH13a] (dated 29 Oct. 2012),* the GGH construction was thought to provide secure SubM, DLIN, and GXDH instantiation. It was soon realized that these problems could be broken in polynomial-time. Therefore, we only introduce these problems and analysis for the CLT version in Section 4.3

*It can be accessed from the IACR eprint server.

Chapter 3

Multilinear maps over the Ideal Lattices and Its Analysis

In this chapter, we introduce GGH multilinear maps and its analysis. The GGH scheme are provided relying on ideal lattices. Basically, its underlying set is a polynomial quotient ring $R = \mathbb{Z}[X]/\langle X^n + 1\rangle$ with a power of two n. Let $\langle \mathbf{g} \rangle \subset R$ be an principal ideal generated by a secret short element $\mathbf{g} \in R$, and $R_q := R/\langle q \cdot R \rangle$ for a fixed large integer q. Then a message space and a ciphertext space are defined by $R/\langle \mathbf{g} \rangle$ and R_q, respectively.

Because the GGH schme is constructed by a graded encoding scheme, it provide a zero-testing parameter and its encodings are defined with level. In GGH scheme, the level-t encoding of $\mathbf{m} \in R$ and zerotesting parameter are of the form:

$$\mathsf{enc}_t(\mathbf{m}) = \frac{\mathbf{r} \cdot \mathbf{g} + \mathbf{m}}{\mathbf{z^t}} \bmod q, \mathbf{p}_{zt} = \frac{\mathbf{h} \cdot \mathbf{z}^\kappa}{\mathbf{g}} \bmod q$$

where $\mathbf{r}$ is a short random element and $\mathbf{z}, \mathbf{h}$ are fixed secret elements. In the original GGH paper, the authors observe that if one can find a short element of $\langle \mathbf{g} \rangle$, the GGH scheme becomes insecure. Therefore in this chapter, we focus on how to recover the short element of $\langle \mathbf{g} \rangle$ from public parameters and use it to break the GGH scheme.

3.1 GGH13 Multilinear maps

First, we briefly recall the Garg *et al.* GGH construction. We refer to the original paper [GGH13a] for a complete description. The scheme relies on the following parameters.

λ: the security parameter

κ: the multilinearity parameter

σ: the basic Gaussian parameter for drawing the ideal generator $\mathbf{g}$

σ': the Gaussian parameter for sampling level-zero elements

σ^*: the Gaussian parameter for constructing nonzero level elements

m: the number of encodings of zero in the public parameters at each level.

q: the modulus of a ciphertext

n: the dimension of a base ring

Instance generation: $(\mathsf{params}, \mathbf{p}_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^\kappa)$.
For a given λ and κ, $\mathsf{InstGen}(1^\lambda, 1^\kappa)$ outputs the following elements.

Sample $\mathbf{g} \leftarrow \mathcal{D}_{R,\sigma}$ until $\|\mathbf{g}\|, \|\mathbf{g}^{-1}\| \leqslant n^2$ and, $\mathcal{I} = \langle \mathbf{g} \rangle$ is a prime ideal in R.
Sample $\mathbf{z} \leftarrow_R R_q$.
Sample $\mathbf{a} \leftarrow \mathcal{D}_{1+\mathcal{I},\sigma'}$ and set a level-1 encoding of 1, $\mathbf{y} = \left[\frac{\mathbf{a}}{\mathbf{z}}\right]_\mathbf{q}$.
Sample $X_i = \{\mathbf{b}_{ij}\mathbf{g}\} \leftarrow \mathcal{D}_{\mathcal{I},\sigma'}$ and set a level-i encoding of zero, $\mathbf{x}_{ij} = \left[\frac{\mathbf{b}_{ij}\mathbf{g}}{\mathbf{z}^\mathbf{i}}\right]_q$ for all $1 \leqslant i \leqslant \kappa, 1 \leqslant j \leqslant m$.
Sample $\mathbf{h} \leftarrow \mathcal{D}_{R,\sqrt{q}}$ and set a zero-testing parameter $\mathbf{p}_{zt} = \left[\frac{\mathbf{h}}{\mathbf{g}}\mathbf{z}^\kappa\right]_\mathbf{q}$.

Then publish $\mathsf{params} = (n, q, \kappa, \mathbf{y}, \{\mathbf{x}_{ij}\})$ and $\mathbf{p}_{zt}$.

Sampling level-zero encodings: $\mathbf{a} \leftarrow \mathsf{samp}(\mathsf{params})$.
To sample a valid level-0 encoding $\mathbf{a}$, we draw an element from $D_{\mathcal{I},\sigma'}$. In other words, $\mathbf{a} \leftarrow D_{\mathcal{I},\sigma'}$.

Encodings at higher levels: $\mathbf{c}_i \leftarrow \mathsf{enc}(\mathsf{params}, i, \mathbf{c})$.
Given a level-j encoding $\mathbf{c}$ for $j < i$, we draw an m integers $r_k \leftarrow \mathcal{D}_{\mathbb{Z},\sigma*}$ $1 \leqslant k \leqslant m$ and compute $\mathbf{c}_i = \left[\mathbf{c} \cdot \mathbf{y}^{i-j} + \sum_{k=1}^{m} r_k \cdot \mathbf{x}_{ik}\right]_q$.

Adding and multiplying encodings:
Given two encodings $\mathbf{c}_1$ and $\mathbf{c}_2$ of the same level, the sum of $\mathbf{c}_1$ and $\mathbf{c}_2$ is computed by $\mathsf{Add}(\mathbf{c}_1, \mathbf{c}_2)=[\mathbf{c}_1 + \mathbf{c}_2]_q$. Given two encodings $\mathbf{c}_1$ and $\mathbf{c}_2$, we multiply $\mathbf{c}_1$ and $\mathbf{c}_2$ by $\mathsf{Mul}(\mathbf{c}_1, \mathbf{c}_2)=[\mathbf{c}_1 \cdot \mathbf{c}_2]_q$.

Zero testing: $\mathsf{isZero}(\mathsf{params}, \mathbf{p}_{zt}, \mathbf{c}) \stackrel{?}{=} 0/1$.
To test given a level-κ encoding $\mathbf{c}$ is zero or not, we compute $\|[\mathbf{p}_{zt} \cdot \mathbf{c}]_q\|_\infty$. So zerotesting procedure returns 1 if $\|[\mathbf{p}_{zt} \cdot \mathbf{c}]_q\|_\infty < q^{3/4}$; otherwise, return 0.

Extraction: $v \leftarrow \mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, \mathbf{c})$.
To extract a canonical representation of a coset from an encoding $\mathbf{u} = [\mathbf{c}/\mathbf{z}^\kappa]_q$, we just multiply by the zero-testing parameter $\mathbf{p_{zt}}$, collect the $(\log q)/4\grave{}\,\lambda$ most-significant bits of each of the n coefficients of the result. In other words, for given a level-κ encoding $\mathbf{c}$, compute $MSB_{\log q/4-\lambda}([\mathbf{p}_{zt} \cdot \mathbf{c}]_q)$.

This works because for any two encodings $\mathbf{u}, \mathbf{u'}$ of the same coset we have

$$\|\mathbf{p}_{zt} \cdot \mathbf{u} - \mathbf{p}_{zt} \cdot \mathbf{u'}\| = \|\mathbf{p}_{zt} \cdot (\mathbf{u} - \mathbf{u'})\| < q^{3/4},$$

therefore we guarantee $\mathbf{p}_{zt} \cdot \mathbf{u}$, $\mathbf{p}_{zt} \cdot \mathbf{u'}$ to agree on their $(\log q)/4\grave{}\,\lambda$ most significant bits without negligible probability.

Hardness Assumptions

We recall the definitions of the graded decisional Diffie-Hellman problem (GDDH) for GGH scheme and its computational variant GCDH. They do not seem to be reducible to more classical assumptions in generic ways.

For an adversary $\mathcal{A}$ and the parameters λ and κ, we consider the following process in the GGH scheme.

1. Choose $(n, q, \kappa,\ \{\mathbf{x}_{\kappa j}\},\ \mathbf{p}_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^\kappa)$.
2. Sample $\mathbf{m}_i \leftarrow \mathsf{samp}(\mathsf{params})$ for each $0 \leqslant i \leqslant \kappa$.
3. Set $\mathbf{u}_i = \left[\dfrac{\mathbf{r_i} \cdot \mathbf{g} + \mathbf{m_i}}{\mathbf{z}}\right]_q \leftarrow \mathsf{enc}(\mathsf{params},\ 1,\ \mathbf{m_i})$ for all $0 \leqslant i \leqslant \kappa$.
4. Choose $\mathbf{r} \leftarrow D_{R,\sigma'}$.
5. Sample $\rho_j \leftarrow \{0, 1\}$ for $1 \leqslant j \leqslant m$.
6. Set $\hat{\mathbf{u}} = \left[\mathbf{m}_0 \times \prod_{i=1}^{\kappa} \mathbf{u}_i + \sum_j \rho_j \mathbf{x}_{\kappa j}\right]_q$.
7. Set $\mathbf{u} = \left[\mathbf{r} \times \prod_{i=1}^{\kappa} \mathbf{u}_i + \sum_j \rho_j \mathbf{x}_{\kappa j}\right]_q$.

The GDDH problem is to distinguish between two distributions, $\mathcal{D}_{DDH}$ and $\mathcal{D}_R$, where

$$\mathcal{D}_{DDH} = \{q, \{\mathbf{x}_i\}, \mathbf{p}_{zt}, \mathbf{u}_0, \ldots, \mathbf{u}_\kappa, \hat{\mathbf{u}}\} \text{ and } \mathcal{D}_R = \{q, \{\mathbf{x}_i\}, \mathbf{p}_{zt}, \mathbf{u}_0, \ldots, \mathbf{u}_\kappa, \mathbf{u}\}.$$

The GCDH problem is to output a level-κ encoding of $\prod_{i=0}^{\kappa} \mathbf{m}_i + \mathcal{I}$ given the inputs

$$\{q, \{\mathbf{x}_i\}, \mathbf{p}_{zt}, \mathbf{u}_0, \ldots, \mathbf{u}_\kappa\}.$$

Parameter Settings.

Garg *et al.* suggested to set the parameters so that the following conditions are met:

- $\sigma = \sqrt{\lambda n}$, $\sigma' = \lambda n^{3/2}$: to bound the size of $\mathbf{g}$ and level-1 encodings. It means that with overwhelming probability $\|\mathbf{g}\| \leqslant n\sqrt{\lambda}$ and $\|\mathbf{a}\|, \|\mathbf{b_{ij}}\| \leqslant$

λn^2.

- $\sigma^* = 2^\lambda$, $m = O(n^2)$: so that the distribution of level-1 encoding $\mathbf{u}$ is independent to its re-randomized one of $\mathbf{u}'$.
- $q \geqslant 2^{8\kappa\lambda} \cdot n^{O(\kappa)}$: to valid zero test, we need to $q^{1/8}$ larger than the size of denominator of level-κ encodings.
- $n = \tilde{O}(\kappa\lambda^2)$: to thwart lattice reduction attacks.

3.2 Basic Notions

In this section, we introduce basic notions related to analysis of GGH scheme.

Lattices.

An m-dimensional lattice of rank n is the set of all integer combinations $\{\sum_{i=1}^{n} x_i \mathbf{b_i} \mid x_i \in \mathbb{Z}\}$ of n linear independent vectors $\mathbf{b_1}, \ldots, \mathbf{b_n} \in \mathbb{R}^{\mathbf{m}}$. The set of $\mathbf{b_i}$'s is called a basis of L. If $n = m$, L is called a full rank lattice. A sublattice is a subset $L' \subset L$ that is also a lattice. A matrix $\mathbf{B} = [\mathbf{b_1} \cdots \mathbf{b_n}]$ is a basis matrix of a lattice L. If $L \subset \mathbb{Z}^n$, we call it an integral lattice. In this study, we only focus on integral lattices.

Given a lattice L in $\mathcal{R}^m$ with a basis matrix $\mathbf{B}$, the determinant of the lattice is defined as $\sqrt{\det(\mathbf{B^T B})}$, where $\mathbf{B^T}$ is the transpose matrix of $\mathbf{B}$ and $\det(M)$ denotes the determinant of a square matrix M. By abusing the notations, $\det L$ is used to denote the determinant of L. The i-th successive minimum of a lattice L, denoted by $\lambda_i(L)$, is defined by

$$\lambda_i(L) = \min_r \{r : \dim(\mathsf{span}(L \cap B(r))) \geqslant i\},$$

where $B(r) : \{x \in \mathbb{R}^n : \|x\| \leqslant r\}$.

Given a basis $\{\mathbf{b_1}, \cdots, \mathbf{b_n}\}$, we denote by $\mathbf{b_1^*}, \cdots, \mathbf{b_n^*}$ the output of the Gram-Schmidt orthogonalization of the basis, i.e., $\mathbf{b_i^*}$ is the component of $\mathbf{b_i}$ orthogonal to $\mathrm{span}(\mathbf{b_1}, \cdots, \mathbf{b_{i-1}})$.

Ideal lattice.

For an element $\mathbf{g} \in R$, we denote the principal ideal in R generated by $\mathbf{g}$ by $\langle \mathbf{g} \rangle$, whose basis consists of $\{\mathbf{g}, X\mathbf{g}, \ldots, X^{n-1}\mathbf{g}\}$. By identifying a polynomial $\mathbf{g} = \sum g_i X^i \in R$ with a vector $(g_{n-1}, g_{n-2}, \ldots, g_0)^T$ in $\mathbb{Z}^n$, we can apply lattice theory to the algebraic ring R and algebraic ring theory to the ideal lattice $\langle \mathbf{g} \rangle$. For a polynomial $\mathbf{u} \in R$ and a basis $\mathcal{B} := \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$, we denote the reduction of $\mathbf{u}$ modulo the fundamental region of lattice $\mathcal{B}$ by $\mathbf{u} \bmod \mathcal{B}$; that is, $\mathbf{u} \bmod \mathcal{B}$ is the unique representation of $\mathbf{u} \in R$ such that $\mathbf{u} - (\mathbf{u} \bmod \mathcal{B}) \in \mathcal{B}$ and $\mathbf{u} \bmod \mathcal{B} = \sum_{i=0}^{n-1} \alpha_i \mathbf{b}_i$ for $\alpha_i \in (-1/2, 1/2]$. For the polynomials $\mathbf{u}, \mathbf{v} \in R$, we use the notation $\mathbf{u} \bmod \mathbf{v}$ as $\mathbf{u} \bmod \mathcal{V}$, where $\mathcal{V}$ is a basis $\{\mathbf{v}, X\mathbf{v}, \ldots, X^{n-1}\mathbf{v}\}$. By the definition of $\mathbf{u} \bmod \mathbf{v}$, it is of the form $\sum_{i=0}^{n-1} \alpha_i X^i \mathbf{v}$ for $\alpha_i \in (-1/2, 1/2]$. Hence, the size of its Euclidean norm is bounded by $\sum_{i=0}^{n-1} \|X^i \mathbf{v}\|/2 = \sum_{i=0}^{n-1} \|\mathbf{v}\|/2 = \frac{n}{2}\|\mathbf{v}\|$.

Lattice reduction algorithm

To find a short element of a lattice, lattice reduction algorithms such as the LLL algorithm and the BKZ algorithm are described in [LLL82, HPS11, ADRSD14]. These algorithms lead us to find an approximately short vector of a lattice with bounded time.

In the case of the BKZ algorithm, by [HPS11], the quality of the algorithms relies on the block size β. More precisely, using the BKZ algorithm upon basis $\mathbf{B} = \{\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_n}\}$, we obtain a reduced basis $\mathbf{B'} = \{\mathbf{b'_1}, \mathbf{b'_2}, \cdots, \mathbf{b'_n}\}$, which satisfies:

- $\|\mathbf{b'}_1\| \leqslant 2(\gamma_\beta)^{\frac{n-1}{2(\beta-1)}+\frac{3}{2}} \cdot (\det \mathcal{L})^{\frac{1}{n}}$ in $\mathsf{poly}(n, size(\mathbf{B})) \cdot \mathcal{C}_{\mathbf{HKZ}}(\beta)$ times or
- $\|\mathbf{b'}_1\| \leqslant 4(\gamma_\beta)^{\frac{n-1}{\beta-1}+3} \cdot (\Lambda_1(\mathcal{L}))^{\frac{1}{n}}$ in $\mathsf{poly}(n, size(\mathbf{B})) \cdot \mathcal{C}_{\mathbf{HKZ}}(\beta)$ times,

where $\mathcal{L}$ is the lattice $\mathcal{L}(\mathbf{B})$, $\gamma_\beta \leqslant \beta$ is the Hermite constant of rank β, $size(\mathbf{B})$ is the size of the largest entries of the basis matrix $\mathbf{B}$, and $\mathcal{C}_{HKZ}(\beta) = 2^{O(\beta)}$ is the cost of HKZ-reduction in dimension β. For an output $\mathbf{b'_1}$ of lattice

reduction algorithms, we call $\frac{\mathbf{b_1'}}{\det(\mathcal{L})^{1/n}}$ and $\frac{\mathbf{b_1'}}{\lambda_1(\mathcal{L})}$ as a Hermite factor and approximate factor, respectively.

For convenience of calculation, throughout this paper, we assume that we have a lattice reduction algorithm $\mathcal{A}_\delta$, whose output contains a short vector $\mathbf{b}$ with Euclidean norm less than $\delta^n \cdot \det(\mathcal{L})^{1/n}$ or $\delta^{2n} \cdot \lambda_1(\mathcal{L})$ for an n-dimensional lattice $\mathcal{L}$ instead of $2(\gamma_\beta)^{\frac{n-1}{2(\beta-1)}+\frac{3}{2}} \cdot (\det \mathcal{L})^{\frac{1}{n}}$ or $4(\gamma_\beta)^{\frac{n-1}{\beta-1}+3} \cdot (\Lambda_1(\mathcal{L}))^{\frac{1}{n}}$, respectively.

3.3 Attack on GGH with low level encodings of zero

In this section, we explain an attack algorithm, which is a different approach from [HJ16], to solve the GDDH problem of the GGH scheme with low-level encodings of zero

Now we are given a set

$$\{n,\ q,\ \kappa,\ \mathbf{y},\ \{\mathbf{x}_{ij}\}, \mathbf{p}_{zt}, \mathbf{u}_0 = \mathsf{enc}_1(\mathbf{m_0}), \ldots, \mathbf{u}_\kappa = \mathsf{enc}_1(\mathbf{m}_\kappa), \mathbf{w}\},$$

where $\mathbf{w}$ is a challenge element and it can be written of the form $\mathsf{enc}_\kappa(\mathbf{r} \cdot \prod_{i=1}^{\kappa} \mathbf{m_i})$. Then our attack algorithm consists of the following four steps:

- First, find a basis of an ideal lattice $\langle \mathbf{g} \rangle$.
- Second, compute a short element $\mathbf{d} \cdot \mathbf{g} \in \langle \mathbf{g} \rangle$
- Third, recover an short element $\mathbf{r'}$ from $\mathbf{w}$ and $\mathbf{d} \cdot \mathbf{g}$ such that $\mathbf{r'} - \mathbf{r} \in \langle \mathbf{g} \rangle$
- Last, determine whether an element $\mathbf{r'} \cdot \mathbf{y} - \mathbf{u_0}$ is an encoding of zero or not.

Except for the second step, the whole process consists of easy calculations. Therefore it implies that the GDDH problem is reduced to SVP on ideal lattice.

At first when a short element in $\langle \mathbf{g} \rangle$ is given, we describe an cryptanalysis which is introduced in original GGH paper. Using different κ-products of level-1 encoding of one $\mathbf{y}$ and level-1 encoding of zero $\mathbf{x_{1j}}$'s, one can generate several level-κ encodings of zero and its zerotesting value is of the form:

$$[\mathbf{y}^i \cdot \prod_{k=1}^{\kappa-i} \mathbf{x}_{1j_k} \cdot \mathbf{p}_{zt}]_q = \mathbf{a}^i \cdot \prod_{k=1}^{\kappa-i} \mathbf{b}_{1j_k} \cdot \mathbf{g}^{\kappa-i-1} \cdot \mathbf{h} \tag{3.3.1}$$

Here, essential property is two fold. The first one is that the right hand side of the equation 3.3.1 is not reduced modulo q, because it is a zerotesting value of level-κ encoding of zero, its size is smaller than $q^{3/4}$. The second one is that all of the quantity has the multiple of $\mathbf{h}$. It implies that it is an element of ideal lattice $\langle \mathbf{h} \rangle$. Hence, by obtaining these several elements, one can recover a basis of the ideal lattice $\langle \mathbf{h} \rangle$.

With a similar arguments, when $\kappa - i - 1 \geqslant 1$, the right hand side is a multiple of $\mathbf{g} \cdot \mathbf{h}$ and one can recover a basis of the ideal lattice $\langle \mathbf{g} \cdot \mathbf{h} \rangle$. Using the basis of $\langle \mathbf{h} \rangle$ and $\langle \mathbf{g} \cdot \mathbf{h} \rangle$, we can also recover the basis of $\langle \mathbf{g} \rangle$.

Suppose we have an element of $\langle \mathbf{g} \rangle$ smaller than $q^{1/4}$. Then any element of $\langle \mathbf{g} \rangle$ has the form $\mathbf{d} \cdot \mathbf{g}$ for some $\mathbf{d} \leqslant \mathbf{q^{1/4}}$. By multiplying it to zerotesting parameter, we make a modified zerotesting parameter $\mathbf{p}'_{zt} = [\mathbf{d} \cdot \mathbf{g} \cdot \mathbf{h} \cdot \mathbf{z}^\kappa / \mathbf{g}]_\mathbf{q} = [\mathbf{d} \cdot \mathbf{h} \cdot \mathbf{z}^\kappa]_\mathbf{q}$. Here we assume $\mathbf{w}$ is a right product of $\mathbf{u_i}$. Next we multiply the modified zerotesting parameter to $\mathbf{w}$ and $\prod_{i=1}^{\kappa} \mathbf{u_i}$, respectively. Because these are level-κ encodings, it can be written as $\mathbf{w} = \dfrac{\mathbf{c_1} \cdot \mathbf{g} + \prod_{\mathbf{i=0}}^{\kappa} \mathbf{m_i}}{\mathbf{z}^\kappa} \bmod \mathbf{q}$ and $\prod_{i=1}^{\kappa} \mathbf{u_i} = \dfrac{\mathbf{c_2} \cdot \mathbf{g} + \prod_{\mathbf{i=1}}^{\kappa} \mathbf{m_i}}{\mathbf{z}^\kappa} \bmod \mathbf{q}$ for some small $\mathbf{c_i} \in R$. Then we have following quantities:

$$\begin{aligned} [\mathbf{p}'_{zt} \cdot \mathbf{w}]_q &= \mathbf{d} \cdot \mathbf{h} \cdot \left(\mathbf{c_1} \cdot \mathbf{g} + \prod_{i=0}^{\kappa} \mathbf{m_i} \right) \\ \text{and } [\mathbf{p}'_{zt} \cdot \prod_{i=1}^{\kappa} \mathbf{u}_i]_q &= \mathbf{d} \cdot \mathbf{h} \cdot \left(\mathbf{c_2} \cdot \mathbf{g} + \prod_{i=1}^{\kappa} \mathbf{m_i} \right). \end{aligned}$$

Since, both of the right hand side are smaller than $q^{1/4} \cdot q^{1/2} \cdot q^{1/8} <$

$q/2$, there is no modulus reduction for q. Hence, it satisfies $[\mathbf{p}'_{zt} \cdot \mathbf{w}]_q / [\mathbf{p}'_{zt} \cdot \prod_{i=1}^{\kappa} \mathbf{u}_i]_q = \mathbf{m}_0 \bmod \langle \mathbf{g} \rangle$. Because we know the basis of $\langle \mathbf{g} \rangle$, we can compute the quantity easily. Finally, we can reduce the value modulo the rotation basis $\{\mathbf{d} \cdot \mathbf{g}, \cdots, \mathbf{d} \cdot \mathbf{g} \cdot X^{n-1}\}$. This yields a short element $\mathbf{m}'_0$ and it is a valid level-0 encoding of $\mathbf{m}_0$. Hence, the zerotesting value $[(\mathbf{m}'_0 \cdot \mathbf{y} - \mathbf{u}_0) \cdot \mathbf{y}^{\kappa-1} \cdot \mathbf{p}_{zt}]_q$ are smaller than $q^{3/4}$. If $\mathbf{w}$ is not a right product, the above value cannot be small with overwhelming probability. Hence, one can solve the GDDH problems.

Since the overall attack algorithm consists of only basic arithmetics, the time complexity of the algorithm heavily relies on finding short vector of $\langle \mathbf{g} \rangle$.

We now how to find such a short vector in the lattice. Actually, a desired short element in the ideal lattice can be computed in quasi-polynomial time in λ, which yields a break of the GGH scheme with the current parameters.

3.3.1 Sublattice Algorithm

Currently, all approximate lattice reduction algorithms have exponentially large Hermite factors in the rank of the input matrix. For example, LLL has a Hermite factor $2^{n/4}$ for the rank n of input matrix L. To obtain a short vector with a smaller Hermite factor, we may attempt to apply LLL to a sublattice of L of smaller dimension. However, it is not always true that a sublattice of smaller dimension has a determinant smaller than that of the original lattice L.

In this subsection, we show how to use HNF to obtain a sublattice having a determinant not larger than that of the original lattice.

Theorem 3.3.1. *Given a basis matrix* $[\mathbf{b_1}, \cdots, \mathbf{b_n}]$ *of integral lattice* L *in Hermite normal form,* $\det([\mathbf{b_i}, \cdots, \mathbf{b_n}]) \geqslant \det([\mathbf{b_{i+1}}, \cdots, \mathbf{b_n}])$ *for* $1 \leqslant i \leqslant n-1$.

Proof. Suppose $\mathbf{b_i}$ is a vector with pivot d_i. Consider an ordered basis $\mathbf{B_i} := \{\mathbf{b_n}, \cdots, \mathbf{b_i}\}$ and its corresponding Gram-Schmidt basis $\{\mathbf{b_n^*}, \cdots, \mathbf{b_i^*}\}$. Then $\det(\mathbf{B_i}) = \prod_{\mathbf{j=i}}^{\mathbf{n}} \|\mathbf{b_j^*}\| = \|\mathbf{b_i^*}\| \cdot \det(\mathbf{B_{i+1}})$. Here $\mathbf{b_i^*}$ is of the form $\mathbf{b_i} - (\mathbf{c_n b_n} + \cdots + \mathbf{c_{i+1} b_{i+1}})$ for some $c_n, \ldots, c_{i+1} \in \mathcal{R}$ and so the first nonzero component

of $\mathbf{b_i^*}$ is also d_i. Hence, $\|\mathbf{b_i^*}\| \geqslant \mathbf{d_i}$ and $\det(\mathbf{B_i}) \geqslant \mathbf{d_i} \cdot \det(\mathbf{B_{i+1}})$. Since d_i is a positive integer by the definition of HNF, we have $\det(\mathbf{B_i}) \geqslant \det(\mathbf{B_{i+1}})$.

□

An Algorithm for SVP

We apply the LLL algorithm to a sublattice L' of integral lattice L with $\det(L') \leqslant \det(L)$ obtained as in the previous section. It is described in Algorithm 1.

Algorithm 1 LLL with HNF

Input: $\{\mathbf{b_1}, \cdots, \mathbf{b_n}\}$ and $m < n$
Output: $\{\mathbf{v_1}, \ldots, \mathbf{v_m}\}$
1. Compute the HNF basis $(\mathbf{b'_1}, \cdots, \mathbf{b'_n})$ of $(\mathbf{b_1}, \cdots, \mathbf{b_n})$.
2. Set $(\mathbf{v_1}, \ldots, \mathbf{v_m})$ as the output vector of LLL upon input
$(\mathbf{b'_{n-m+1}}, \cdots, \mathbf{b'_n})$
return $(\mathbf{v_1}, \ldots, \mathbf{v_m})$.

If we take an appropriate m, the first vector of the output of **Algorithm 1** is shorter than that of LLL on the original lattice when the determinant of the lattice is not large.

Theorem 3.3.2. *For an n-dimensional integral lattice L with $\log\det(L) < n^2/4$ and $m = \left\lfloor 2\sqrt{\log(\det(L))} \right\rceil$,* **Algorithm 1** *outputs an LLL-reduced basis of a certain sublattice L' of L with $\det L' \leqslant \det L$ in polynomial time in n and $\det(L)$. In particular, the first vector satisfies $\mathbf{v}_1 \in L$ such that $\|\mathbf{v}_1\| < 2^{\sqrt{\log\det(L)+1/2}}$.*

Proof. Let $\{\mathbf{b'_1}, \cdots, \mathbf{b'_n}\}$ be an HNF basis of lattice L. By the given conditions, $m = \left\lfloor 2\sqrt{\log(\det(L))} \right\rceil < n$. When Algorithm 1 is applied to L, the output is merely the output of LLL on the sublattice L' generated by $\{\mathbf{b'_{n-m+1}}, \cdots, \mathbf{b'_n}\}$.

Moreover, since $\det(L') \leqslant \det(L)$ by Theorem 1, the first vector $\mathbf{v_1}$ of the output satisfies

$$\|\mathbf{v}_1\| \leqslant 2^{m/4} \cdot \det(L')^{1/m} \leqslant 2^{m/4} \cdot \det(L)^{1/m} = 2^{\sqrt{\log(\det(L))+\epsilon}},$$

where $0 \leqslant \epsilon \leqslant 1/2$.

Obviously, $\mathbf{v_1} \in \mathbf{L'} \subset \mathbf{L}$, and the running time is upper bounded by the running time of LLL and HNF. □

Another lattice reduction algorithm, the BKZ algorithm with block size β, outputs a lattice vector of a size bounded by $2(\gamma_\beta)^{\frac{n-1}{2(\beta-1)}+\frac{3}{2}} \cdot (\det L)^{\frac{1}{n}}$ in $\mathsf{poly}(n, \mathsf{size}(\mathbf{B})) \cdot \mathcal{C}_{\mathbf{HKZ}}(\beta)$ times, where γ_β is the Hermite constant of rank β, $\mathsf{size}(\mathbf{B})$ is the size of largest entries of basis matrix $\mathbf{B}$ of L, and $\mathcal{C}_{HKZ}(\beta) = 2^{O(\beta)}$ is the cost of HKZ-reduction in dimension β [HPS11, ADRSD14].

We claim that our algorithm asymptotically outperforms BKZ with block size $\beta = \mathsf{polylog}(n)$ for an n-dimensional integral lattice with determinant 2^{n^δ} for $\delta < 2$. While the output vector of BKZ with β has a Euclidean norm of at most $2^{(n \log \beta)/\beta + O(1)}$, **Algorithm 1** with BKZ gives a vector of the Euclidean norm at most $2^{n^{\delta/2}}$, which is asymptotically smaller than the previous one. We remark that when the basis entries of an n-dimensional lattice L are bounded by $2^{o(n)}$, we achieve a subquadratic determinant $2^{o(n^2)}$ of L.

Furthermore, we may plug BKZ in our algorithm (instead of LLL) to obtain a shorter lattice vector $\mathbf{b} \in L$. More precisely, when $\det L = \beta^{o(n^2)}$, we have $\|\mathbf{b}\| < 2\beta^{\sqrt{(2 \log_\beta \det(L))/(\beta-1)}+\frac{3}{2}}$.The running time is upper bounded by the running time of the BKZ algorithm with block size β, $\mathsf{poly}(n, size(\mathbf{B})) \cdot \mathcal{C}_{\mathbf{HKZ}}(\beta)$, and the running time of computing HNF.

As an application of our algorithm, we find a short vector of an ideal lattice $\langle \mathbf{g} \rangle$ used in GGH scheme. The determinant of the ideal lattice $\langle \mathbf{g} \rangle$ is equal to $\det[\mathbf{g}, x\mathbf{g}, \cdots, x^{n-1}\mathbf{g}]$, which is bounded by $\|\mathbf{g}\|^n$.

Applying the **Algorithm 1** with β-block size BKZ , we can have the following result, a variant of **Theorem 3.3.2**:

Theorem 3.3.3. *Given an element* $\mathbf{g}$ *of* $R = \mathbb{Z}[x]/\langle x^n + 1 \rangle$ *with* $\|\mathbf{g}\| <$

$\beta^{n/(2\beta)}$, *one can find an element* $\mathbf{f} \in \langle \mathbf{g} \rangle$ *such that* $\|\mathbf{f}\| \leqslant 2\beta\sqrt{(2\log_\beta \det(L))/(\beta-1)}+3/2 = 2^{O\left(\sqrt{\frac{n\log\beta\log\|\mathbf{g}\|}{\beta}}\right)}$ *in* $\mathsf{poly}(n, \log\|\mathbf{g}\|) \cdot \mathcal{C}_{HKZ}(\beta)$ *running times.*

According to previous arguments in Section 3.3 if one finds a short element $\mathbf{d} \cdot \mathbf{g}$ in $\langle \mathbf{g} \rangle$ such that $\|\mathbf{d} \cdot \mathbf{g}\| \leqslant \mathbf{q^{1/4}}$, the GGH scheme is not secure. Since a basis of the ideal lattice $\langle \mathbf{g} \rangle$ is easily recovered in polynomial time, we may apply our **Algorithm 1** to find a short element in the lattice $\langle \mathbf{g} \rangle$ and break the GGH scheme. From the **Theorem 3.3.3**, we can obtain an element $\mathbf{d} \cdot \mathbf{g} \in \langle \mathbf{g} \rangle$ satisfying

$$\|\mathbf{d} \cdot \mathbf{g}\| \leqslant 2^{O\left(\sqrt{\frac{n\log\beta\log\|\mathbf{g}\|}{\beta}}\right)} = 2^{O\left(\lambda\sqrt{\frac{\log\beta\log\lambda}{\beta}}\right)},$$

which is asymptotically smaller than $q^{1/4}$ for $\beta = \log^2(\lambda)$. Hence, it yields that GDDH problem of the GGH scheme with the current parameters can be broken in quasi-polynomial time in λ.

3.4 Attack on GGH with top level encodings of zero

Since some applications such as indistinguishable obfuscation have no low level encodings of zero, the analysis of the GDDH with low level encoding of zero is not enough to explain the hardness of GGH scheme. In this section, we provide an algorithm for the GDDH problem of the GGH scheme only using top level encodings of zero. More precisely, suppose we have

$$\{n,\ q,\ \kappa,\ \mathbf{y},\ \{\mathbf{x}_{\kappa i}\},\ \mathbf{p}_{zt},\ \mathbf{u}_0, \cdots, \mathbf{u}_\kappa, \mathbf{w}\}.$$

If one can find a basis of $\langle \mathbf{g} \rangle$, by sublattice algorithm one can also recover a short element $\mathbf{d} \cdot \mathbf{g} \in \langle \mathbf{g} \rangle$. It leads to generate a low level encoding of zero by multiplying the short element to level-1 encoding of one $\mathbf{y}$. Hence, as an attack algorithm suggested in Section 3.3, one can solve the GDDH problems.

Now we propose an idea to obtain a basis of $\langle \mathbf{g} \rangle$. Considering $[\mathbf{u}_1^\kappa/\mathbf{x}_{\kappa 1}]_q =$

$[\mathbf{m}_1^\kappa/\mathbf{b}_{\kappa 1}\mathbf{g}]_q$, the sizes of the denominator and numerator are bounded by $n^{3.5\kappa}\sqrt{n}^{\kappa-1} < n^{4\kappa}$ and $n^{3.5}$, which are very smaller than modulus q. In other words, it is written by ratio of small polynomial in ring R_q. If we recover the denominator and numerator, one can obtain the multilple of $\mathbf{g}$. By repeating the procedure, it gives several multiples of $\mathbf{g}$ and a basis of $\mathbf{g}$. This means that GDDH is reduced to the NTRU problem because it is precisely the NTRU problem of finding elements from small ratio. Especially, the size of q is exponential of λ, it corresponds to ONTRU problem.

3.4.1 Overstretched NTRU Problem and Its Analysis

In this section, we now explain the formal definition of NTRU problem and how to solve it using lattcie reduction algorithms. First of all, we state an NTRU problem as follows:

Definition 3.4.1 (The NTRU problem)**.**
Let n and q be integers, M be a positive real number and $F(X)$ be a degree n integral polynomial. For a polynomial ring $R := \mathbb{Z}[X]/\langle F(X)\rangle$, $\mathbf{f}$ and $\mathbf{g}$ are sampled from R and have Euclidean norms bounded by M. For given a polynomial $\mathbf{h} = [\mathbf{f}/\mathbf{g}]_q$, the NTRU problem $\mathsf{NTRU}_{R,q,M,\tau}$ is to find $\mathbf{a}\cdot\mathbf{f}, \mathbf{a}\cdot\mathbf{g} \in R$ for some $\mathbf{a} \in \mathbf{R}$, such that $\|\mathbf{a}\cdot\mathbf{f}\|, \|\mathbf{a}\cdot\mathbf{g}\| \leqslant \tau$.

In many NTRU-based applications, M is taken to be similar to $\mathsf{poly}(n)$. Furthermore, when q is set to be super-polynomial in n, the NTRU problem is called the *overstretched* NTRU problem (ONTRU). In [ABD16, KF17], the authors solved ONTRU problem on $R = \mathbb{Z}[X]/\langle X^n + 1\rangle$ for a power of two n with $\tau = q$. In this paper we focus on the *O*NTRU problem on $R = \mathbb{Z}[X]/\langle X^n + 1\rangle$ with $\tau = q/2$ and $M = \mathsf{poly}(n)$.

Lattice Based Algorithms for NTRU

Now, we state an useful lemma to solve the NTRU problem.

Lemma 3.4.1 (**[GGH13a], Lemma 3).** *Let* $\mathbf{f}, \mathbf{g} \in \mathbf{R} = \mathbb{Z}[\mathbf{X}]/\langle \mathbf{X^n} + \mathbf{1} \rangle$ *be relative prime and* $[\mathbf{g}]_\mathbf{q}$ *is invertible in* $[R]_q = R/qR$*. If* $\mathbf{c} \in R$ *satisfies* $\|\mathbf{c}\| < \dfrac{q}{2\sqrt{n} \cdot \|\mathbf{f}\|}$ *and* $\|[\mathbf{c} \cdot \mathbf{f} \cdot \mathbf{g}^{-1}]_q\| < \dfrac{q}{2\sqrt{n} \cdot \|\mathbf{g}\|}$*, then* $\mathbf{c}$ *and* $[\mathbf{c} \cdot \mathbf{f} \cdot \mathbf{g}^{-1}]_q$ *are contained in the ideal* $\langle \mathbf{g} \rangle$ *and* $\langle \mathbf{f} \rangle$*, respectively.*

Proof. Let $\mathbf{w} := [\mathbf{c} \cdot \mathbf{f} \cdot \mathbf{g}^{-1}]_q$. Then, $[\mathbf{gw}]_q = [\mathbf{cf}]_q$. Since $\|\mathbf{w}\| < q/(2\|\mathbf{g}\| \cdot \sqrt{n})$, we have $\|\mathbf{gw}\| \leqslant \|\mathbf{g}\| \cdot \|\mathbf{w}\| \cdot \sqrt{n} \leqslant q/2$ and $\|\mathbf{cf}\| \leqslant \|\mathbf{c}\| \cdot \|\mathbf{f}\| \cdot \sqrt{n} \leqslant q/2$. Therefore, $\mathbf{gw} = \mathbf{cf}$ in $\mathbb{Z}[X]/\langle X^n + 1 \rangle$. Because $\mathbf{cf} \in \langle \mathbf{g} \rangle$ and $\mathbf{f}$ is a relative prime to $\mathbf{g}$, we can conclude $\mathbf{c} \in \langle \mathbf{g} \rangle$. With similar reasons, we have $\mathbf{w} \in \langle \mathbf{f} \rangle$. □

Hence, **Lemma 3.4.1** gives if one can find a short pair $([\mathbf{c} \cdot \mathbf{h}]_\mathbf{q}, \mathbf{c})$ with an Euclidean norm smaller than $\dfrac{q}{2 \cdot \sqrt{n} \cdot M}$, one will be able to solve the $\mathsf{NTRU}_{R,q,M,\frac{q}{2}}$.

By employing the above property, we can describe the basic lattice-based approach to solve the NTRU problem with an input polynomial $\mathbf{h} = [\mathbf{f}/\mathbf{g}]_\mathbf{q}$. Let consider the lattice $\mathcal{L}$ generated by following $2n \times 2n$ basis matrix:

$$\mathbf{B} = \begin{pmatrix} q \cdot \mathbf{I}_n & \phi(\mathbf{h}) \\ \mathbf{O} & \mathbf{I}_n \end{pmatrix} \in \mathbb{Z}^{2n \times 2n}.$$

For a polynomial $\mathbf{c} = \sum_{\mathbf{i=0}}^{\mathbf{n-1}} \mathbf{c_i} \cdot \mathbf{X^i}$, the polynomial vector $(\mathbf{c} \cdot \mathbf{h}, \mathbf{c}) = \sum_{\mathbf{i=0}}^{\mathbf{n-1}} \mathbf{c_i} \cdot (\mathbf{X^i} \cdot \mathbf{h}, \mathbf{X^i})$ corresponds to a lattice point $\sum_{i=0}^{n-1} c_i \cdot B_{n+i}$, where B_{n+i} is the $n+i$-th column vector of the basis matrix $\mathbf{B}$. It implies that $([\mathbf{c} \cdot \mathbf{h}]_\mathbf{q}, \mathbf{c})$ is also identified to lattice point of $\mathcal{L}(\mathbf{B})$. Hence, finding a short pair $([\mathbf{c} \cdot \mathbf{h}]_\mathbf{q}, \mathbf{c})$ is the same as finding a short lattice point of $\mathcal{L}(\mathbf{B})$.

We also state another lemma to solve NTRU problems. This is applicable to the pair $(\mathbf{a}, \mathbf{b})$ where $\mathbf{b}$ is known to be a multiple of $\mathbf{g}$.

Lemma 3.4.2. *Let* $\mathbf{g}$ *be an element of* $R = \mathbb{Z}[X]/\langle X^n + 1 \rangle$ *and* $\mathbf{f} \in R$ *be relative prime to* $\mathbf{g}$*. For some* $\mathbf{d} \in \mathbf{R}$*, if* $\mathbf{d} \cdot \mathbf{g} \in \langle \mathbf{g} \rangle \subset \mathbf{R}$ *satisfies* $\|\mathbf{d} \cdot \mathbf{g}\| < \dfrac{\mathbf{q}}{\mathbf{2} \cdot \mathbf{n} \cdot \|\mathbf{f}\| \cdot \|\mathbf{g^{-1}}\|}$ *then* $\mathbf{d} \cdot \mathbf{g}$ *and* $[\mathbf{d} \cdot \mathbf{g} \cdot \mathbf{h}]_\mathbf{q}$ *are solution of* $\mathsf{NTRU}_{R,q,M,\frac{q}{2}}$ *problem with input* $\mathbf{h}$*.*

Proof. By conditions, we have

$$\|\mathbf{d}\cdot\mathbf{f}\| \;=\; \|\mathbf{d}\cdot\mathbf{f}\cdot\mathbf{g^{-1}}\cdot\mathbf{g}\| \leqslant \mathbf{C_F^2}\|\mathbf{d}\cdot\mathbf{g}\|\cdot\|\mathbf{f}\|\cdot\|\mathbf{g^{-1}}\| < \mathbf{q/2}.$$

Hence, $[\mathbf{d}\cdot\mathbf{g}\cdot\mathbf{h}]_{\mathbf{q}}$ has the form $\mathbf{d}\cdot\mathbf{f}$. □

Subfield Algorithm

Generally, the dimension is too large to solve the NTRU problem by using a lattice reduction algorithm. Therefore we focus on redcuing the dimension of the NTRU problems. As one of the methods, we discuss how the NTRU problem with a given input $[\mathbf{f}/\mathbf{g}]_q$ is reduced to the NTRU problem with an input whose denominator and numerator have half of the degree of $\mathbf{f}$ and $\mathbf{g}$ using a subfield technique. Concurrently and independently, in [ABD16], the authors introduced the this technique using a Norm map. Instaed of the Norm maps, we exploit the Trace map. In our work, we can achieve the same, but slightly better, results by using the trace map.

Throughout this section, let $n = 2^s$ and denote $\mathbb{Q}[X^{2^t}]/\langle X^n+1\rangle$ and $\mathbb{Z}[X^{2^t}]/\langle X^n+1\rangle$ by K_t and R_t, respectively, with $0 \leqslant t \leqslant s$. Note that $K_s := \mathbb{Q} \leqslant K_{s-1} \leqslant \cdots \leqslant K_0 = \mathbb{Q}[X]/\langle X^n+1\rangle$, where $A \leqslant B$ denotes that A is a subfield of B. Since K_0 is a Galois extension field of K_1 with a degree of 2, $\mathrm{Gal}(K_0/K_1)$ is a group of order 2. That is, $\mathrm{Gal}(K_0/K_1) = \{id, \sigma\}$, satisfying $\sigma(X) = -X$; therefore, $\sigma^2 = id$, where id is the identity map. For an element $\mathbf{h}, \mathbf{g} \in R \subset K_0$, the following elements are contained in $R_1 \subset K_1$:

$$\begin{aligned} Tr_{K_0/K_1}(\mathbf{h}) &= \mathbf{h} + \sigma(\mathbf{h}), \\ N_{K_0/K_1}(\mathbf{h}) &= \mathbf{h}\cdot\sigma(\mathbf{h}), \\ Tr_{K_0/K_t}(\mathbf{h}\sigma(\mathbf{g})) &= \mathbf{h}\sigma(\mathbf{g}) + \sigma(\mathbf{h})\mathbf{g}, \end{aligned}$$

since they are fixed by $\mathrm{Gal}(K_0/K_1)$. Note that these elements have only $n/2$ terms, and the last one lies in $2\cdot R_1$. Generally, for $0 < t \leqslant s$, K_0 is a Galois extension field of K_t with a degree of 2^t and the Galois group

$G_t := \mathrm{Gal}(K_0/K_t) = \{\sigma_0 = id, \sigma_1, \ldots, \sigma_{2^t-1}\}$. For an element $\mathbf{h}, \mathbf{g} \in R \subset K_0$, the following elements are contained in $R_t \subset K_t$:

$$\sum_{i=0}^{2^t-1} \sigma_i(\mathbf{h}) = \mathbf{h} + \sigma_{\mathbf{1}}(\mathbf{h}) + \cdots + \sigma_{\mathbf{2^t-1}}(\mathbf{h}),$$
$$\prod_{i=0}^{2^t-1} \sigma_i(\mathbf{h}) = \mathbf{h} \cdot \sigma_{\mathbf{1}}(\mathbf{h}) \cdot \cdots \cdot \sigma_{\mathbf{2^t-1}}(\mathbf{h}),$$

$$Tr_{K_0/K_t}(\mathbf{h}\sigma_{\mathbf{1}}(\mathbf{g})\sigma_{\mathbf{2}}(\mathbf{g}) \cdots \sigma_{\mathbf{2^t-1}}(\mathbf{g})),$$

since they are fixed by $\mathrm{Gal}(K_0/K_t)$. Moreover, these elements have only $n/2^t$ terms, and the last one lies in $2^t \cdot R_t$. Using this property, we can obtain the following main theorem of this section.

Theorem 3.4.1. *Let q and $m \in \mathbb{Z}$ be integers and let D and N be positive real numbers. Set $B = \min\{\frac{q}{2D\sqrt{n}}, \frac{q}{2N\sqrt{n}}\}$. Then, for $F_n(X) = X^n + 1$ with $n = 2^s$ and $0 < t \leqslant s$, we can reduce $\mathsf{NTRU}_{F_n,q,D,N,B}$ to $\mathsf{NTRU}_{F_{n/2^t},q,D_t,N_t,B_t}$, where $B_t = \min\{\frac{q}{2D_t\sqrt{n}}, \frac{q}{2N_t\sqrt{n}}, \frac{q}{2nN^2\|\mathbf{g}^{-1}\|\sqrt{n}}\}$, $D_t = D^{2^t} \prod_{j=1}^{t} \sqrt{n/2^j}$, and $N_t = ND^{2^t-1} \prod_{j=1}^{t} \sqrt{n/2^j}$.*

Proof. Suppose we are given $[\mathbf{f}/\mathbf{g}]_q$, where $\mathbf{g}$ and $\mathbf{f}$ are sampled from the set $\{(\mathbf{g}, \mathbf{f}) \in R^2 = (\mathbb{Z}[X]/\langle F_n(X)\rangle)^2 : \|\mathbf{f}\| < N, \|\mathbf{g}\| < D\}$. We consider the useful element

$$\begin{aligned} Tr_{K_0/K_t}\left(\frac{\mathbf{f}}{\mathbf{g}}\right) &= \frac{\mathbf{f}}{\mathbf{g}} + \sigma_1\left(\frac{\mathbf{f}}{\mathbf{g}}\right) + \cdots + \sigma_{2^t-1}\left(\frac{\mathbf{f}}{\mathbf{g}}\right) \\ &= \frac{Tr_{K_0/K_t}(\mathbf{f}\sigma_{\mathbf{1}}(\mathbf{g})\sigma_{\mathbf{2}}(\mathbf{g}) \cdots \sigma_{\mathbf{2^t-1}}(\mathbf{g}))}{\prod_{i=0}^{2^t-1} \sigma_i(\mathbf{g})} \\ \left[Tr_{K_0/K_t}\left(\frac{\mathbf{f}}{\mathbf{g}}\right)\right]_q &= \left[\frac{Tr_{K_0/K_t}(\mathbf{f}\sigma_{\mathbf{1}}(\mathbf{g})\sigma_{\mathbf{2}}(\mathbf{g}) \cdots \sigma_{\mathbf{2^t-1}}(\mathbf{g}))}{\prod_{i=0}^{2^t-1} \sigma_i(\mathbf{g})}\right]_q \end{aligned}$$

in K_t and R_q. For the sake of simplicity, we denote a quantity $\left[Tr_{K_0/K_t}\left(\frac{\mathbf{f}}{\mathbf{g}}\right)\right]_q$,

its denominator polynomial, and its numerator polynomial by $\mathbf{h}_t$, $\mathbf{DP}_t$, and $\mathbf{NP}_t$, respectively. Then it satisfies

- $\mathbf{DP}_t \in R_t$, and $\mathbf{NP}_t \in 2^t \cdot R_t$,
- $\left\| \dfrac{\mathbf{NP}_t}{2^t} \right\| \leqslant N D^{2^t-1} \prod_{j=1}^{t} \sqrt{n/2^j}$,
- $\|\mathbf{DP}_t\| \leqslant D^{2^t} \prod_{j=1}^{t} \sqrt{n/2^j}$.

Therefore, we get a new instance $\mathbf{h}_t$ for $\mathsf{NTRU}_{F_{n/2^t},q,D_t,N_t,B_t}$ from an original instance $\mathbf{h}$, where $D_t = D^{2^t} \prod_{j=1}^{t} \sqrt{n/2^j}$, $N_t = N D^{2^t-1} \prod_{j=1}^{t} \sqrt{n/2^j}$, and $B_t = \min\{\frac{q}{2D_t\sqrt{n}}, \frac{q}{2N_t\sqrt{n}}, \frac{q}{2nN^2\|\mathbf{g}^{-1}\|\sqrt{n}}\}$. Now, suppose that a solution $(\mathbf{a}_t, \mathbf{b}_t) \in R_t$ of $\mathsf{NTRU}_{\phi_{n/2^t},q,D_t,N_t,B_t}$ is known such that

$$[\mathbf{b}_t/\mathbf{a}_t]_q = \mathbf{h}_t.$$

Moreover, since $\mathbf{g}$ and $\mathbf{f}$ are relative primes with a high probability [ABD16], we assume the coprimality of $\mathbf{g}$ and $\mathbf{f}$. Then, by **Lemma 3.4.1**, $\mathbf{a}_t$ is of the form $\mathbf{a}_t = \mathbf{d} \cdot \mathbf{DP_t} = \mathbf{d} \cdot \prod_{\mathbf{i=0}}^{\mathbf{2^t-1}} \sigma_{\mathbf{i}}(\mathbf{g})$. After computing

$$[\mathbf{a}_t \cdot \mathbf{h}]_q = \left[\mathbf{d} \prod_{\mathbf{i=0}}^{\mathbf{2^t-1}} \sigma_{\mathbf{i}}(\mathbf{g}) \cdot [\mathbf{f}/\mathbf{g}]_{\mathbf{q}} \right]_q = \left[\mathbf{df} \prod_{i=1}^{2^t-1} \sigma_i(\mathbf{g}) \right]_q,$$

set $\mathbf{a} = \mathbf{a}_t$ and $\mathbf{b} = \left[\mathbf{df} \prod_{\mathbf{i=1}}^{\mathbf{2^t-1}} \sigma_{\mathbf{i}}(\mathbf{g}) \right]_q$. Then, we can conclude that the pair

$(\mathbf{a}, \mathbf{b})$ is a solution of $\mathsf{NTRU}_{\phi_n,q,D,N,B}$ with following properties:

$$\begin{aligned} [\mathbf{b}/\mathbf{a}]_q &= [\mathbf{f}/\mathbf{g}]_q, \\ \|\mathbf{a}\| &\leqslant \frac{q}{2N_t\sqrt{n}} \leqslant \frac{q}{2N\sqrt{n}}, \end{aligned}$$

$$\begin{aligned} \left\|\mathbf{df}\prod_{\mathbf{i=1}}^{\mathbf{2^t-1}}\sigma_{\mathbf{i}}(\mathbf{g})\right\| &= \left\|\mathbf{dg^{-1}f}\prod_{\mathbf{i=0}}^{\mathbf{2^t-1}}\sigma_{\mathbf{i}}(\mathbf{g})\right\| \leqslant \|\mathbf{a}_t\|\cdot\|\mathbf{g}^{-1}\|\cdot\|\mathbf{f}\|\cdot n \\ &< \frac{q}{2nN^2\|\mathbf{g}^{-1}\|\sqrt{n}}\cdot\|\mathbf{g}^{-1}\|\cdot N\cdot n \\ &= \frac{q}{2N\sqrt{n}}. \end{aligned}$$

The last inequality implies that $\mathbf{b} = \left[\mathbf{df}\prod_{\mathbf{i=1}}^{\mathbf{2^t-1}}\sigma_{\mathbf{i}}(\mathbf{g})\right]_q$ is actually $\mathbf{b} = \mathbf{df}\prod_{\mathbf{i=1}}^{\mathbf{2^t-1}}\sigma_{\mathbf{i}}(\mathbf{g})$ in R. Thus, we obtain the desired result. □

Comparing with [ABD16], our result works better when $N \geqslant D$ because the value of our N_1 is smaller than that of [ABD16] while the values of D_1 are same. By **Theorem 3.4.1**, one can solve the ONTRU problem with smaller dimension and obtain the following result.

Theorem 3.4.2. *Let q be an integer, n a power of 2, and λ the security parameter. Let $\mathbf{h} = [\mathbf{f}/\mathbf{g}]_q$ be an instance of the $\mathsf{NTRU}_{\phi_n,q,D,N,B}$ problem with the parameters $\log q = c_1\cdot\lambda^\ell$, $n \leqslant c_2\cdot\lambda^{2\ell}$, $N = q^a$, $0 < a < 1/2$, $D = \lambda^k < N$, $\phi_n(X) = X^n + 1$, and $B = \min\{\frac{q}{2D\sqrt{n}}, \frac{q}{2N\sqrt{n}}\}$. For $\beta > 0$ and $t \in \mathbb{Z}$, if*

$$2\beta^{\frac{n_t}{2(\beta-1)}+\frac{3}{2}}\sqrt{q} \leqslant \min\{\frac{q}{2D_t\sqrt{n}}, \frac{q}{2N_t\sqrt{n}}, \frac{q}{2nN^2\|\mathbf{g}^{-1}\|\sqrt{n}}\},$$

where $D_t = D^{2^t}\prod_{j=1}^{t}\sqrt{n/2^j}$, $N_t = ND^{2^t-1}\prod_{j=1}^{t}\sqrt{n/2^j}$, and $n_t = \frac{n}{2^t}$, then the problem is solved in $2^{O(\beta)}$ time.

In particular, if $\|\mathbf{g}^{-1}\| \leqslant \frac{D^{2^t-1}}{N}\cdot\sqrt{n}^{t-2}\cdot 2^{\frac{t(t+1)}{4}}$ and $\beta = \log^2\lambda$, the problem is solved in $2^{O(\log^2\lambda)}$ time. *

*If $\mathbf{h}$ and $\mathbf{g}$ are sampled from continuous spherical Gaussian distributions, we can obtain a bound of $\|\mathbf{g^{-1}}\|$ with a high probability. [ABD16, Lemma 3]

For example, when $n = \lambda^2$, $D = \lambda^2$, $N = q^{1/8}$, and $\log q = \lambda$, one can solve $\mathsf{NTRU}_{\phi_n,q,D,N,B}$ in quasi-polynomial time in λ.

Proof. By **Theorem 3.4.1**, one can obtain a new instance $[\mathrm{Tr}_{K/K_t}([\mathbf{f}/\mathbf{g}]_q)/2^t]_q \in [R]_q \cap R_t$ for $\mathsf{NTRU}_{\phi_{n_t},q,N_t,D_t,B_t}$. Now, we consider the following column lattice $\mathbf{B}_t$:

$$\mathbf{B}_t = \begin{pmatrix} \mathbf{I}_{n_t} & \phi_t(\mathbf{h}_t) \\ \mathbf{O} & q \cdot \mathbf{I}_{n_t} \end{pmatrix},$$

where I_{n_t} is the identity matrix with a size $n_t = n/2^t$, and $\phi_t(\mathbf{h}_t) \in \mathbb{Z}^{n_t \times n_t}$ is a matrix whose i-th column is $\iota(X^{i2^t}[\mathrm{Tr}_{K/K_t}([\mathbf{f}/\mathbf{g}]_q/2^t]_q))$ for $0 \leqslant i < n/2^t$. In other words, for $[\mathrm{Tr}_{K/K_t}([\mathbf{f}/\mathbf{g}]_q)/2^t]_q = \sum_{j=0}^{n_t-1} h_j X^{j2^t}$, the i-th column of $\phi_t(\mathbf{h}_t)$ is of the form $(-h_{n_t-i}, \cdots, -h_{n_t-1}, h_0, \cdots h_{n_t-i-1})^T$. Using the BKZ algorithm with a block size β, one can obtain an element in $\mathbf{B}_t$,

$$\mathbf{u}_t = (u_0, \cdots, u_{n_t-1}, u_{n_t}, \cdots, u_{2n_t-1})^T,$$

with $\|\mathbf{u}_t\| \leqslant 2\beta^{\frac{n_t-1}{2(\beta-1)}+\frac{3}{2}} \det(\mathbf{B}_t)^{\frac{1}{2n_t}} = 2\beta^{\frac{n_t-1}{2(\beta-1)}+\frac{3}{2}}\sqrt{q}$ [HPS11]. Taking $\mathbf{c} = \sum_{i=0}^{n_t-1} u_i X^{i2^t} \in \mathbb{Z}[X^{2^t}]/\langle X^n+1\rangle$, we then have $[\mathbf{c} \cdot [\mathrm{Tr}_{K/K_t}([\mathbf{f}/\mathbf{g}]_q)/2^t]_q]_q = \sum_{i=0}^{n_t-1} u_{n_t+i} X^{i2^t} \in \mathbb{Z}[X^{2^t}]/\langle X^n+1\rangle$. Moreover, if we choose t such that

$$2\beta^{\frac{n_t}{2(\beta-1)}+\frac{3}{2}}\sqrt{q} \leqslant B_t, \tag{3.4.2}$$

then $\|\mathbf{c}\|$ and $\|[\mathbf{c} \cdot \mathrm{Tr}_{K/K_t}([\mathbf{f}/\mathbf{g}]_q)]_q\|$ satisfy

$$\begin{aligned} \|\mathbf{c}\| &< \|\mathbf{u}_t\| \leqslant 2\beta^{\frac{n_t-1}{2(\beta-1)}+\frac{3}{2}}\sqrt{q} \leqslant B_t \leqslant \frac{q}{2N_t\sqrt{n}}, \\ \|[\mathbf{c} \cdot \mathrm{Tr}_{K/K_t}([\mathbf{f}/\mathbf{g}]_q)]_q\| &< \|\mathbf{u}_t\| \leqslant 2\beta^{\frac{n_t-1}{2(\beta-1)}+\frac{3}{2}}\sqrt{q} \leqslant B_t \leqslant \frac{q}{2D_t\sqrt{n}}. \end{aligned}$$

In other words, $\mathbf{c}$ satisfies the conditions of **Lemma 3.4.1**. Therefore, $\mathbf{c}$ is in $\langle \mathrm{N}_{K/K_t}(\mathbf{g})\rangle \subset \langle \mathbf{g}\rangle$. Note that $\mathbf{c}$ is of the form $\mathbf{c} = \mathbf{d} \cdot \mathrm{N}_{\mathbf{K/K_t}}(\mathbf{g}) = \mathbf{d'g} \in \mathbf{R_t}$ for some $\mathbf{d}$, $\mathbf{d\prime} \in \mathbf{R}$. Hence, by Theorem 1, a pair $(\mathbf{c}, [\mathbf{c} \cdot \mathbf{h}]_\mathbf{q})$ is a solution of $\mathsf{NTRU}_{\phi_n,q,N,D,B}$. The running time of this procedure is dominated by that of

the BKZ algorithm with a block size β, which is $\mathsf{poly}(n, \log q) \cdot \mathcal{C}_{HKZ}(\beta)$ time, where $\mathcal{C}_{HKZ}(\beta) = 2^{O(\beta)}$ is the cost of the HKZ reduction in the dimension β [ADRSD14, HPS11]. When $\|\mathbf{g^{-1}}\| \leqslant \dfrac{\mathbf{D^{2^t-1}}}{\mathbf{N}} \cdot \sqrt{\mathbf{n}}^{\mathbf{t-2}} \cdot \mathbf{2}^{\frac{\mathbf{t(t+1)}}{\mathbf{4}}}$, we obtain $B_t = \frac{q}{2N_t\sqrt{n}}$. To check that the above condition for β and t is satisfied, we have the following equivalence equation:

$$\begin{aligned} & 2\beta^{\frac{n_t}{2(\beta-1)}+\frac{3}{2}}\sqrt{q} \leqslant \frac{q}{2N_t\sqrt{n}} \\ \Leftrightarrow \quad & \left(\frac{n_t}{2(\beta-1)}+\frac{3}{2}\right)\log\beta + \log D_t - \log D + \frac{\log n}{2} + 2 < \frac{\log q}{2} - \log N. \end{aligned}$$

To optimize the left-hand side of the inequality, we choose t such that

$$t = \left\lceil \log \sqrt{\frac{n\log\beta}{2k(\beta-1)\log\lambda}} \right\rceil.$$

Then, the left-hand side is asymptotic to the following:

$$\begin{aligned} & \left(\frac{n_t}{2(\beta-1)}+\frac{3}{2}\right)\log\beta + \log D_t - \log D + \frac{\log n}{2} + 2 \\ \approx \quad & \frac{n}{2^t\cdot 2(\beta-1)}\log\beta + 2^t\log\lambda^k + O(1) \\ \approx \quad & 2\sqrt{\frac{n\log\beta\log\lambda^k}{2(\beta-1)}} + O(1), \end{aligned}$$

where the last approximation originates from the arithmetic–geometric mean. This implies that if one chooses $\beta = \log^2\lambda$, then the last value is asymptotically smaller than $(1/2-a)\log q$. Hence, one can obtain the results. □

Hybrid Algorithm

In this section, we describe an improved algorithm to solve the ONTRU problems upon $\mathbf{h} = [\mathbf{f}/\mathbf{g}]_\mathbf{q}$. We also denote $K_t = \mathbb{Q}[X^{2^t}]/\langle X^n+1\rangle$, $R_t = \mathbb{Z}[X^{2^t}]/\langle X^n+1\rangle$, and $n_t = n/2^t$ as in subfield algorithm.

The subfield attack [ABD16, CJL16] uses only one polynomial $\mathrm{N}_{K/K_t}(\mathbf{h})$ or $\mathrm{Tr}_{K/K_t}(\mathbf{h})$. Instead, we use several polynomials rather than a single one.

Then we have the following theorem.

Theorem 3.4.3. *Let n be a power of two and $\mathbf{g}$ an element of $R = \mathbb{Z}[X]/\langle X^n + 1\rangle$ with square free algebraic norm and $\mathbf{f} \in R$ a relatively prime to $\mathbf{g}$. When the sizes of M and $\|\mathbf{g^{-1}}\|_{\mathbf{K}}$ are $\mathsf{poly}(n)$, and q is super-polynomial in n, one can solve the $\mathsf{NTRU}_{R,q,M,q/2}$ problems upon $[\mathbf{f}/\mathbf{g}]_{\mathbf{q}}$ using the BKZ algorithm with block size β with $\beta/\log\beta \geqslant \frac{27n\log M}{2\log^2 q} + o\left(\frac{n\log n\log M}{\log^3 q}\right)$ in $\mathsf{poly}(n)\cdot 2^{O(\beta)}$ time.*

Generally, it is expected that it would be easier to recover $\mathbf{g}$, if several NTRU instances $\mathbf{h_i} = [\mathbf{f_i}/\mathbf{g}]_{\mathbf{q}}$ are given instead of one element. Unfortunately, there is no algorithm that uses multiple instances to the best of our knowledge.

When an ONTRU instance $[\mathbf{f}/\mathbf{g}]_{\mathbf{q}}$ with square free norm $\mathrm{N}(\mathbf{g})$ is given, we provide a new algorithm that employs several polynomials in the form of $\mathrm{Tr}(\mathbf{h}\cdot\mathbf{X^{-i}})$. More precisely, we prove that if there exist polynomials $\mathbf{c_i} \in \mathbf{R_t}$ such that the size of $\{\mathbf{c_i}\}_{\mathbf{i=0}}^{\mathbf{n_t-1}}$ and $\sum_{i=0}^{n_t-1} \mathbf{c_i}\cdot\mathrm{Tr}(\mathbf{h}\cdot\mathbf{X^{-i}})/\mathbf{2^t}$ is small, $\sum_{i=0}^{n_t-1} \mathbf{c_i}\cdot\mathbf{X^{-i}}$ is a multiple of $\mathbf{g}$.[†] The proof reduces the NTRU problem to finding a short vector in an appropriate lattice. By applying a sublattice algorithm, we can get the above theorem.

Now we start proving the **Theorem 3.4.3**. Let $\mu_{j,t}(\mathbf{a}) = \dfrac{\mathrm{Tr}_{K/K_t}(\mathbf{a}\cdot\mathbf{X^{-j}})}{2^t}$ for given t. Then $\mathbf{a} = \sum_{i=0}^{n-1} a_i\cdot X^t \in R$ could be expressed as $\sum_{j=0}^{2^t-1} \mu_{j,t}(\mathbf{a})\cdot X^j$. Using a μ notation, the $\mathbf{h}$ is of the form $\sum_{j=0}^{2^t-1} \mu_{j,t}(\mathbf{h})\cdot X^j$. Our strategy is to use several polynomials $\mu_{0,t}(\mathbf{h}),\ \mu_{\mathbf{1},\mathbf{t}}(\mathbf{h}),\ \cdots,\ \mu_{\mathbf{m-1},\mathbf{t}}(\mathbf{h})$ rather than a single one. From now on, we use μ_j instead of $\mu_{j,t}$ for simplicity. Then, we can have the following lemma, extended from **Lemma 3.4.1**.

Lemma 3.4.1. *Let q be an integer and n be a power of two and $n_t = n/2^t$, $\mathbf{f},\mathbf{g} \in R = \mathbb{Z}[X]/\langle X^n+1\rangle$ and $\mathbf{g}$ has a square free norm $\mathrm{N}_{K/\mathbb{Q}}(\mathbf{g})$, $\mathbf{h} = [\mathbf{f}/\mathbf{g}]_{\mathbf{q}} \in \mathbf{R_q}$. If $\mathbf{c_i} \in R_t = \mathbb{Z}[X^{2^t}]/\langle X^n+1\rangle$ for $0 \leqslant i < 2^t$ satisfies the*

[†]In particular if $\mathbf{c_i}$ is equal to zero for $1 \leqslant i \leqslant n_t - 1$, it is an original subfield attack.

following inequalities:

$$\|\mathbf{c_i}\| < \frac{q}{2^{t+1} \cdot C_F^3 \cdot \|\mathbf{f}\| \cdot \|\mathbf{g^{-1}}\|_{\mathbf{K}} \cdot \|\mathbf{g}\|^{\mathbf{2^t}}} \text{ for all } i$$
$$\left\| \left[\sum_{i=0}^{2^t-1} \mathbf{c_i} \cdot \mu_i(\mathbf{h}) \right]_q \right\| < \frac{q}{2C_F \cdot \|\mathbf{g}\|^{2^t}},$$

then $\mathbf{c} = \sum_{\mathbf{i=0}}^{\mathbf{2^t-1}} \mathbf{c_i} \cdot \mathbf{X^{-i}}$ *is contained in the ideal* $\langle \mathbf{g} \rangle$ *, and the pair* $(\mathbf{c},\ [\mathbf{c} \cdot \mathbf{f} \cdot \mathbf{g^{-1}}]_{\mathbf{q}})$ *is a solution of* $\mathsf{NTRU}_{R,q,M,q/2}$.

Proof. Throughout in this proof, we write $\mathrm{Tr}(\mathbf{a})$ and $\mathrm{N}(\mathbf{a})$ instead of $\mathrm{Tr}_{K/K_t}(\mathbf{a})$ and $\mathrm{N}_{K/Kt}(\mathbf{a})$. and define $\tilde{\mathbf{h}} = \mathbf{f} \cdot \mathbf{g^{-1}} \in \mathbf{K}$. Trivially, $[\tilde{\mathbf{h}}]_{\mathbf{q}} = \mathbf{h}$. Let $\mathbf{w} := \left[\sum_{\mathbf{i=0}}^{\mathbf{2^t-1}} \mathbf{c_i} \cdot \mu_{\mathbf{i}}(\tilde{\mathbf{h}}) \right]_{\mathbf{q}}$. Since $\mathrm{N}(\mathbf{g}) \in \mathbf{R_t}$, we get $\mathrm{N}(\mathbf{g}) \cdot \mu_{\mathbf{i}}(\tilde{\mathbf{h}}) = \mathrm{N}(\mathbf{g}) \cdot \mathrm{Tr}(\tilde{\mathbf{h}} \cdot \mathbf{X^{-i}})/\mathbf{2^t} = \mathrm{Tr}(\mathrm{N}(\mathbf{g}) \cdot \tilde{\mathbf{h}} \cdot \mathbf{X^{-i}})/\mathbf{2^t} = \mu_{\mathbf{i}}(\mathrm{N}(\mathbf{g}) \cdot \tilde{\mathbf{h}})$, and following equality holds:

$$[\mathrm{N}(\mathbf{g}) \cdot \mathbf{w}]_{\mathbf{q}} = \left[\mathrm{N}(\mathbf{g}) \cdot \sum_{\mathbf{i=0}}^{\mathbf{2^t-1}} \mathbf{c_i} \cdot \mu_{\mathbf{i}}(\tilde{\mathbf{h}}) \right]_{\mathbf{q}} = \left[\sum_{\mathbf{i=0}}^{\mathbf{2^t-1}} \mathbf{c_i} \cdot \mu_{\mathbf{i}}(\tilde{\mathbf{h}} \cdot \mathrm{N}(\mathbf{g})) \right]_{\mathbf{q}}.$$

By two conditions of lemma, we have

1. $\|\mathrm{N}(\mathbf{g}) \cdot \mathbf{w}\| \leqslant \mathbf{C_F} \cdot \|\mathrm{N}(\mathbf{g})\| \cdot \|\mathbf{w}\| \leqslant \mathbf{C_F} \cdot \|\mathbf{g}\|^{\mathbf{2^t}} \cdot \|\mathbf{w}\| \leqslant \mathbf{q/2}$

2. $$\left\| \sum_{i=0}^{2^t-1} \mathbf{c_i} \cdot \mu_i(\tilde{\mathbf{h}} \cdot \mathrm{N}(\mathbf{g})) \right\| \leqslant C_F \cdot \sum_{i=0}^{2^t-1} \|\mathbf{c_i}\| \cdot \|\mu_{\mathbf{i}}(\tilde{\mathbf{h}} \cdot \mathrm{N}(\mathbf{g}))\|$$
$$\leqslant C_F \cdot \sum_{i=0}^{2^t-1} \|\mathbf{c_i}\| \cdot \|\tilde{\mathbf{h}} \cdot \mathrm{N}(\mathbf{g})\| \leqslant \mathbf{C_F^3} \cdot \sum_{\mathbf{i=0}}^{\mathbf{2^t-1}} \|\mathbf{c_i}\| \cdot \|\mathbf{f}\| \cdot \|\mathbf{g^{-1}}\|_{\mathbf{K}} \cdot \|\mathbf{g}\|^{\mathbf{2^t}} \leqslant \mathbf{q/2}$$

Therefore, $\mathrm{N}(\mathbf{g}) \cdot \mathbf{w} = \sum_{\mathbf{i=0}}^{\mathbf{2^t-1}} \mathbf{c_i} \cdot \mu_{\mathbf{i}}(\tilde{\mathbf{h}} \cdot \mathrm{N}(\mathbf{g})) = \sum_{\mathbf{i=0}}^{\mathbf{2^t-1}} \mathbf{c_i} \cdot \mathrm{Tr}(\tilde{\mathbf{h}} \cdot \mathrm{N}(\mathbf{g}) \cdot \mathbf{X^{-i}})/\mathbf{2^t}$

in R_t. It is rewritten as

$$2^t \cdot \mathrm{N}(\mathbf{g}) \cdot \mathbf{w} - \sum_{\mathbf{i=0}}^{\mathbf{2^t-1}} \mathbf{c_i} \cdot \left(\mathrm{Tr}(\mathbf{X^{-i}} \cdot \mathbf{f} \cdot \mathrm{N}(\mathbf{g})/\mathbf{g}) - \mathbf{X^{-i}} \cdot \mathbf{f} \cdot \mathrm{N}(\mathbf{g})/\mathbf{g})\right) = (\sum_{i=0}^{2^t-1} \mathbf{c_i} \cdot X^{-i}) \cdot \mathbf{f} \cdot \mathrm{N}(\mathbf{g})/\mathbf{g}.$$

Because the left hand side of the equation is a multiple of $\mathbf{g}$ and $\mathbf{f} \cdot \mathrm{N}(\mathbf{g})/\mathbf{g}$ is a relative prime to $\mathbf{g}$ by the conditions, we can conclude $\sum_{i=0}^{2^t-1} \mathbf{c_i} \cdot X^{-i} \in \langle \mathbf{g} \rangle$. Moreover, we get

$$\| \sum_{i=0}^{2^t-1} \mathbf{c_i} \cdot \mathbf{X^{-i}} \| \leqslant \sum_{i=0}^{2^t-1} \|\mathbf{c_i}\| < \frac{\mathbf{q}}{\mathbf{2} \cdot \mathbf{C_F^3} \cdot \|\mathbf{f}\| \cdot \|\mathbf{g^{-1}}\|_\mathbf{K} \cdot \|\mathbf{g}\|^{\mathbf{2^t}}} < \frac{\mathbf{q}}{\mathbf{2C_F^2} \cdot \|\mathbf{f}\| \cdot \|\mathbf{g^{-1}}\|_\mathbf{K}}$$

as a condition. Finally, **Lemma 3.4.2** shows that $(\mathbf{c},\ [\mathbf{c} \cdot \mathbf{f} \cdot \mathbf{g^{-1}}]_\mathbf{q})$ is a solution of $\mathsf{NTRU}_{R,q,M,\frac{q}{2}}$. □

Next, we consider the following matrix for $\mathbf{h} = \sum_{\mathbf{i=0}}^{\mathbf{2^t-1}} \mu_\mathbf{i}(\mathbf{h}) \cdot \mathbf{X^i}$ so that we find such a vector $\mathbf{c} \in \mathbf{R}$

$$\widehat{\mathbf{B}}_t = \begin{pmatrix} q \cdot \mathbf{I_{n_t}} & \phi_t(\mu_0(\mathbf{h})) & \phi_t(\mu_1(\mathbf{h})) & \cdots & \phi_t(\mu_{2^t-1}(\mathbf{h})) \\ 0 & \mathbf{I_{n_t}} & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{I_{n_t}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \mathbf{I_{n_t}} \end{pmatrix},$$

where $\phi_t(\mu_i(\mathbf{h}))$ is a basis matrix corresponding to the ideal lattice $\langle \mu_i(\mathbf{h}) \rangle$ over $\mathbb{Z}[X^{n_t}]/\langle X^n + 1 \rangle$. Suppose that one can find a lattice point

$$\mathbf{b} = (\mathbf{b'}||\mathbf{b_0}|| \cdots ||\mathbf{b_{2^t-1}}) \in \mathcal{L}(\hat{\mathbf{B}}_\mathbf{t})$$

such that $\|\mathbf{b}\| \leqslant \frac{\mathbf{q}}{\mathbf{2^{t+1}} \cdot \mathbf{C_F^3} \cdot \|\mathbf{f}\| \cdot \|\mathbf{g^{-1}}\|_\mathbf{K} \cdot \|\mathbf{g}\|^{\mathbf{2^t}}}$. Then, trivially, $\|\mathbf{b_i}\| \leqslant \|\mathbf{b}\|$

and $\left\| \left[\sum_{i=0}^{2^t-1} \mathbf{b_i} \cdot \mu_i(\mathbf{h}) \right]_q \right\| = \|\mathbf{b}'\| \leqslant \|\mathbf{b}\|$. Hence, the short vector of $\mathcal{L}(\widehat{\mathbf{B}}_t)$ guarantee to find a vector satisfying the condition of **Lemma 3.4.1**.

To find a short lattice point, we apply the lattice reduction algorithms A_δ to a sublattice L' generated by the first $n_t + m$ column vector of the matrix $\widehat{\widehat{B}}_t$. Therefore if we have:

$$\delta^{n_t+m} q^{\frac{n_t}{n_t+m}} \leqslant \frac{q}{2^{t+1} \cdot C_F^3 \cdot \|\mathbf{f}\| \cdot \|\mathbf{g^{-1}}\|_{\mathbf{K}} \cdot \|\mathbf{g}\|^{\mathbf{2^t}}},$$

we can find a lattice point $\mathbf{b}$ as we want.

Finally, in order to achieve the optimizing condition, one can get the following inequality by taking the logarithm function on both side. Therefore we have the following asymptotic inequality:

$$(n_t + m) \log \delta + \frac{n}{n_t} \log M + \frac{n_t}{n_t + m} \log q \leqslant \log q + o(\log n),$$

The equation means that $\log \delta \leqslant \frac{\log^2 q}{27n \log M}$ is the condition of δ to solve the NTRU problem. Hence, we can solve $\mathsf{NTRU}_{R,q,M,\frac{q}{2}}$ in $\mathsf{poly}(n) \cdot 2^{O(\beta)}$ time if $\beta / \log \beta \geqslant \frac{27n \log M}{2 \log^2 q} + o\left(\frac{n \log n \log M}{\log^3 q}\right)$, using the BKZ algorithm with block size β.

Application to GGH: Finding a lattice point of $\langle \mathbf{g} \rangle$

Now we explain how to use the **Theorem 3.4.3** in order to recover a basis of $\langle \mathbf{g} \rangle$. Suppose we have top level encodings of zero$\{\mathbf{x}_{\kappa i} = [\mathbf{b}_{\kappa i}\mathbf{g}/\mathbf{z}^\kappa]_q$ and level-1 encodings $\mathbf{u}_i = [\mathbf{r}_i\mathbf{g} + \mathbf{m}_i/\mathbf{z}^\kappa]_q = [\mathbf{m}'_i/\mathbf{z}^\kappa]_q$ $0 \leqslant i \leqslant \kappa$. Note that $\|\mathbf{b}_{\kappa i}\mathbf{g}\|, \|\mathbf{m}'_i\| \leqslant \sigma^* \sqrt{n} \leqslant n^{3.5}$ with overwhelming probability. For convenience, we use the notation G_t to denote $\mathrm{Gal}(K/K_t)$. Considering $[\mathbf{u}_1^\kappa / \mathbf{x}_{\kappa 1}]_q = [\mathbf{m'}_1^\kappa / \mathbf{b}_{\kappa 1}\mathbf{g}]_q$, the sizes of the denominator and numerator are bounded by $n^{3.5\kappa}\sqrt{n}^{\kappa-1} < n^{4\kappa}$ and $n^{3.5}$, respectively. Using the algorithm in **Theorem 3.4.3** for several $[\mathbf{m'}_I / \mathbf{b}_{\kappa \mathbf{j}}\mathbf{g}]_q := [\mathbf{m'}_{i_1} \cdots \mathbf{m'}_{i_\kappa} / \mathbf{b}_{\kappa j}\mathbf{g}]_q$ for $I = [i_1, \cdots, i_\kappa]$,

$i_1, \cdots, i_\kappa \in \{0, \cdots, \kappa\}$, and $j \in \{1, \cdots, m\}$, one can recover several multiples $\mathbf{c_I b'_{\kappa j}} \cdot \mathbf{g'b_{\kappa j}} \cdot \mathbf{g}$ of $\mathrm{N}_{K/K_t}(\mathbf{g})$, where $\mathbf{b'_{\kappa j}} = \prod_{\sigma \in \mathbf{G_t} \setminus \{\mathbf{id}\}} \sigma(\mathbf{b}_{\kappa \mathbf{j}})$ and $\mathbf{g'} = \prod_{\sigma \in \mathbf{G_t} \setminus \{\mathbf{id}\}} \sigma(\mathbf{g})$. Multiplying it by $[\mathbf{m'}_I / \mathbf{b}_{\kappa j}\mathbf{g}]_q$, one can obtain $A_{I,j} = \mathbf{m'}_I \mathbf{c_I b'}_{\kappa \mathbf{j}} \mathbf{g'}$. We remark that $A_{I,j}$ is in $R \backslash R_t$ because $A_{I,j}$ is not fixed for any subgroup of G_t, except the trivial group. Moreover, although $A_{I,j}$ is not in $\langle \mathbf{g} \rangle$, we have $\delta(A_{I,j}) = \delta(\mathbf{m'}_I \mathbf{c_I b'}_{\kappa \mathbf{j}} \mathbf{g'}) = \delta(\mathbf{m'}_I \mathbf{c}_I) \cdot \prod_{\sigma \in G_t \setminus \{\delta\}} \sigma(\mathbf{b}_{\kappa \mathbf{j}} \cdot \mathbf{g}) \in \langle \mathbf{g} \rangle$ for $\delta \in G_t \backslash \{id\}$. One can easily see that $\{\delta(A_{I,j})\}_{\delta \in G_t \backslash \{id\}}$ only have a common factor $\mathbf{g}$. Therefore, using $\{\delta(A_{I,j})\}_{\delta \in G_t \backslash \{id\}}$, we recover a basis matrix of the ideal lattice of $\langle \mathbf{g} \rangle$. Using $\mathrm{N}_{K/K_t}(\mathbf{a})$ for $\mathbf{a} \in \langle \mathbf{g} \rangle$, which is a multiple of $\mathrm{N}_{K/K_t}(\mathbf{g})$, one can also recover a basis matrix of the ideal lattice of $\langle \mathrm{N}_{K/K_t}(\mathbf{g}) \rangle$. Now, using the sublattice algorithm with β block-BKZ algorithm, one can obtain an element $\mathbf{cg} \in \langle \mathrm{N}_{K/K_t}(\mathbf{g}) \rangle$ such that $\|\mathbf{cg}\| \leqslant 2\beta^{\frac{n_t - 1}{2(\beta - 1)} + \frac{3}{2}} \cdot n^{2^{t+1}}$

In summary, we can obtain the following corollary.

Corollary 3.4.3.1. *Given* $\{n, q, \kappa, \{\mathbf{x}_{\kappa i}\}, \mathbf{p}_{zt}, \mathbf{u}_0, \cdots, \mathbf{u}_\kappa, \mathbf{w}\}$ *of the GDDH instances, where* n *is* $\Theta(\lambda^2)$, $\log q = \Theta(\lambda)$, $\mathbf{x}_{\kappa i}$ *is a level-*κ *encoding of zero,* $\mathbf{u}_i$ *is a level-*1 *encoding of* $\mathbf{m}_i$, *and* $\mathbf{w}$ *is a challenge element, one can solve the GDDH in the GGH scheme in* $2^{O(\log^2 \lambda)}$.

According to this Corollary, using the parameters suggested by [GGH13a] leads to attack a security ground of this scheme in the quasi-polynomial time of its security parameter. Thus, n must be at least $\Omega(\lambda^3)$ when $\log q = \Theta(\lambda)$ with the security parameter λ to avoid our attack.

Remark. We assume that the size of the numerator is small for the top level encoding of zero, but in practice this assumption may not be valid for applications. That is, the size of the numerator of the top level encoding of zero is very large, and the GDDH problem in this case is still a hard problem.

Chapter 4

Multilinear Maps over the Integers and Its Analysis

In this chapter, we introduce CLT multilinear maps and its analysis. The CLT scheme are provided relying over the integers and Chinese remainder theorem (CRT). In other words, all elements of CLT are provided in the form of CRT. In this chapter, for n primes $p_1, \ldots, p_n$ we use a notation to express the CRT form

$$\mathsf{CRT}_{(p_1,\ldots,p_n)}(r_1, \ldots, r_n) \text{ or } \mathsf{CRT}_{(p_i)}(r_i),$$

which means an integer congruent to r_i in modulo p_i for all $1 \leqslant i \leqslant n$ in range $[-\prod_{i=1}^{n} p_i/2,\ \prod_{i=1}^{n} p_i/2]$. By CRT, it is uniquely determined.

Basically, its underlying set of CLT is a $\mathbb{Z}^n$. Let $g_i \in \mathbb{Z}$ $1 \leqslant i \leqslant n$ be small integers and p_i, $1 \leqslant i \leqslant n$, large integers. Then a message space and a ciphertext space are defined by $\prod_{i=1}^{n} \mathbb{Z}/\langle g_i \cdot \mathbb{Z}\rangle$ and $\mathbb{Z}/\langle \prod_{i=1}^{n} p_i \cdot \mathbb{Z}\rangle$, respectively.

Because also the CLT schme is constructed by a graded encoding scheme, it provide a zero-testing parameter and its encodings are defined with level. A difference with GGH scheme is that CLT provides n zerotesting parameters in order to sure the zerotesting process in CLT work properly. However, some applications give up on that guarantee and provide only one zerotesting parameter. For that reason, this thesis suggests an analysis using only one zerotesting parameter. In CLT scheme, the level-t encoding of

$\mathbf{m} = (m_1, \cdots, m_n) \in \mathbb{Z}^n$ and zerotesting parameter are of the form:

$$\begin{aligned} \mathsf{enc}_t(\mathbf{m}) &= \frac{r_i \cdot g_i + m_i}{z^t} \bmod p_i, \\ \mathbf{p}_{zt} &= \left[h_i \cdot (z^\kappa \cdot g_i^{-1})\right]_{p_i} \cdot \prod_{i' \neq i} p_{i'} \bmod p_i, \end{aligned}$$

where r_i is a small random integer and z, h are fixed secret integers. If one can recover a secret p_i, the CLT scheme with public data is reduced to GGH scheme over integer. With a similar argument described in Section 3.3, one can recover $\langle h_i \rangle$ and $\langle g_i \rangle$ over $\mathbb{Z}$ so h_i and g_i are also recovered. Independently from $\mathsf{enc}_t(\mathbf{m}_1)/\mathsf{enc}_t(\mathbf{m}_2) = (r_{i1} \cdot g_i + m_{i1})/(r_{i2} \cdot g_i + m_{i2}) \bmod p_i$, one can obtain the its denominator and numerator. Then its modulus reduction with g_i reveals the secret message. It means that to find p_i leads to break the CLT scheme. Hence, we focus on finding a prime p_i.

4.1 The CLT13 Multilinear Map.

First, we briefly recall the Coron *et al.* construction. We refer to the original paper [CLT13] for a complete description. The scheme relies on the following parameters.

λ: the security parameter

κ: the multilinearity parameter

ρ: the bit length of the randomness used for encodings

α: the bit length of the message slots

η: the bit length of the secret primes p_i

n: the number of distinct secret primes

τ: the number of level-1 encodings of zero in public parameters

ℓ: the number of level-0 encodings in public parameters

ν: the bit length of the image of the multilinear map

β: the bit length of the entries of the zero-test matrix H

Instance generation: $(\mathsf{params}, \mathbf{p}_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^\kappa)$. Set the scheme parameters as explained above. For $i \in [n]$, generate η-bit primes p_i, α-bit primes g_i, and compute $x_0 = \prod_{i\in[n]} p_i$. Sample $z \leftarrow \mathbb{Z}_{x_0}$. Let $\Pi = (\pi_{ij}) \in \mathbb{Z}^{n\times n}$ with $\pi_{ij} \leftarrow (n2^\rho, (n+1)2^\rho) \cap \mathbb{Z}$ if $i = j$, otherwise $\pi_{ij} \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z}$. For $i \in [n]$, generate $\vec{r}_i \in \mathbb{Z}^n$ by choosing randomly and independently in the half-open parallelepiped spanned by the columns of the matrix Π and denote by r_{ij} the j-th component of $\vec{r}_i$. Generate $\mathbf{H} = (h_{ij}) \in \mathbb{Z}^{n\times n}$, $\mathbf{A} = (a_{ij}) \in \mathbb{Z}^{n\times \ell}$ such that $\mathbf{H}$ is invertible and $\|\mathbf{H}^T\|_\infty \leqslant 2^\beta$, $\|(\mathbf{H}^{-1})^T\|_\infty \leqslant 2^\beta$ and for $i \in [n]$, $j \in [\ell]$, $a_{ij} \leftarrow [0, g_i)$. Then define:

$$\begin{aligned}
y &= \mathsf{CRT}_{(p_i)}\left(\frac{r_i g_i + 1}{z}\right), \text{ for } i \in [n],\\
x_j &= \mathsf{CRT}_{(p_i)}\left(\frac{r_{ij} g_i}{z}\right) \text{ for } j \in [\tau],\\
x'_j &= \mathsf{CRT}_{(p_i)}(x'_{ij}) = \mathsf{CRT}_{(p_i)}(r'_{ij} g_i + a_{ij}), \text{ for } i \in [n], j \in [\ell],\\
(\mathbf{p}_{zt})_j &= \left[\sum_{i=1}^{n} \left[h_{ij} \cdot (z^\kappa \cdot g_i^{-1})\right]_{p_i} \cdot \prod_{i'\neq i} p_{i'}\right]_{x_0} \text{ for } j \in [n],
\end{aligned}$$

where $r_i \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z}$ and $r'_{ij} \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z}$. Output $\mathsf{params} = (n, \eta, \alpha, \rho, \beta, \tau, \ell, \nu, x_0, y, \{x_j\}, \{x'_j\}, \{\Pi_j\}, s)$ and $\mathbf{p}_{zt}$. Here s is a seed for a strong randomness extractor, which is used for an "Extraction" procedure. We do not recall the latter as it is not needed to describe our attack.

Re-randomizing level-1 encodings: $c' \leftarrow \mathsf{reRand}(\mathsf{params}, c)$. For $j \in [\tau], i \in [n]$, sample $b_j \leftarrow \{0,1\}$, $b'_i \leftarrow [0, 2^\mu) \cap \mathbb{Z}$, with $\mu = \rho + \alpha + \lambda$. Return $c' = [c + \sum_{j\in[\tau]} b_j \cdot x_j + \sum_{i\in[n]} b'_i \cdot \Pi_i]_{x_0}$. Note that this is the only procedure in the CLT multilinear map that uses the x_j's.*

*This procedure can be adapted to higher levels $1 < k \leqslant \kappa$ by publishing appropriate quantities in params.

Adding and multiplying encodings: Given two encodings $\mathbf{c}_1$ and $\mathbf{c}_2$ of the same level, the sum of $\mathbf{c}_1$ and $\mathbf{c}_2$ is computed by $\mathsf{Add}(\mathbf{c}_1, \mathbf{c}_2)=[\mathbf{c}_1 + \mathbf{c}_2]_{x_0}$. Given two encodings $\mathbf{c}_1$ and $\mathbf{c}_2$, we multiply $\mathbf{c}_1$ and $\mathbf{c}_2$ by $\mathsf{Mul}(\mathbf{c}_1, \mathbf{c}_2)=[\mathbf{c}_1 \cdot \mathbf{c}_2]_{x_0}$.

Zero-testing: $\mathsf{isZero}(\mathsf{params}, \mathbf{p}_{zt}, u_\kappa) \stackrel{?}{=} 0/1$. Given a level-$\kappa$ encoding c, return 1 if $\|[\mathbf{p}_{zt} \cdot c]_{x_0}\|_\infty < x_0 \cdot 2^{-\nu}$, and return 0 otherwise.

Coron *et al.* also described a variant where only one such $(\mathbf{p}_{zt})_j$ is given out, rather than n of them (see [CLT13, Se. 6]) and some applications are use the variant. Since, our attack requires only one $(\mathbf{p}_{zt})_j$, without loss of generality, we denote one zerotesting parameter as $\mathbf{p}_{zt}$ without index notation. In [GLW14, App. B.3], Gentry *et al.* described a variant of the above construction that aims at generalizing asymmetric cryptographic bilinear maps. Our attack can be adapted to that variant.

Hardness Assumptions

Instead of recalling the definitions of the GDDH problem of the CLT scheme. We introduce the more difficult problem we are trying to solve. For given the inputs $\mathsf{params} = (n, \eta, \alpha, \rho, \beta, \tau, \ell, \nu, x_0, y, \{x_j\}, \{x'_j\}, \{\Pi_j\}, s)$ and $\mathbf{p}_{zt}$, the hardness problem is to output a prime factor of

Parameter Settings.

Coron *et al.* suggested to set the parameters so that the following conditions are met:

- $\rho = \Omega(\lambda)$: to avoid brute force attack (see also [LS14] for a constant factor improvement).
- $\alpha = \lambda$: so that the ring of messages $\mathbb{Z}_{g_1} \times \ldots \times \mathbb{Z}_{g_n}$ does not contain a small subring $\mathbb{Z}_{g_i}$.[†]

[†]In fact, it seems that making the primes g_i public, equal, and $\Omega(\kappa)$ may not lead to any specific attack [CLT14b].

- $n = \Omega(\eta \cdot \lambda)$: to thwart lattice reduction attacks.
- $\ell \geqslant n \cdot \alpha + 2\lambda$: to be able to apply the leftover hash lemma from [CLT13, Le. 1].
- $\tau \geqslant n \cdot (\rho + \log_2(2n)) + 2\lambda$: to apply leftover hash lemma from [CLT13, Se. 4].
- $\beta = \Omega(\lambda)$: to avoid the so-called gcd attack.
- $\eta \geqslant \rho_\kappa + \alpha + 2\beta + \lambda + 8$, where ρ_κ is the maximum bit size of the random r_i's a level-κ encoding. When computing the product of κ level-1 encodings and an additional level-0 encoding, one obtains $\rho_\kappa = \kappa \cdot (2\alpha + 2\rho + \lambda + 2\log_2 n + 2) + \rho + \log_2 \ell + 1$.
- $\nu = \eta - \beta - \rho_f - \lambda - 3$: to ensure zero-test correctness.

4.2 CRT-ACD with auxiliary input and Its Analysis

In this section, we introduce a CRT-ACD problem with auxiliary input and analyze the problem. Next by adapting the analysis of CRT-ACD with auxiliary input to CLT multilinear maps, we describe to find p_i from params and zerotesting parameter of the CLT scheme.

The approximate greatest common divisor problem (ACD) is initially introduced by Howgrave-Graham [HG01]. It is a problem to find a secret prime p given many near-multiples of p. One of the promising applications of this problem is a homomorphic encryption scheme [vDGHV10]. The scheme has superiority in regard to conceptual simplicity compared to other homomorphic encryption schemes based on lattice problems.

The ACD problem is naturally extended by using multiple primes rather than a single one. An instance of the problem is an integer of the form $p_i q_i + r_i$ for each prime p_i. Therefore, it can be defined by using Chinese

Remainder Theorem (CRT). Now we give a precise definition of an extended ACD problem, which is called CRT-ACD problem.

Definition 4.2.1. (CRT-ACD) Let $n, \eta, \varepsilon \in \mathbb{N}$, and χ_ε be a distribution over $\mathbb{Z} \cap (-2^\varepsilon, 2^\varepsilon)$. For given η bit primes $p_1, \cdots, p_n$, the sampleable CRT-ACD distribution $\mathcal{D}_{\chi_\varepsilon,\eta,n}(p_1, \cdots, p_n)$ is defined as

$$\mathcal{D}_{\chi_\varepsilon,\eta,n}(p_1, \cdots, p_n) = \{\mathsf{CRT}_{(p_i)}(r_i) \mid r_i \leftarrow \chi_\varepsilon\}.$$

The CRT-ACD problem is: For given many samples from $\mathcal{D}_{\chi_\varepsilon,\eta,n}(p_1, \cdots, p_n)$ and $x_0 = \prod_{i=1}^{n} p_i$, find p_i for all i.

Cheon *et al.* gave a batch homomorphic encryption [CCK+13] based on a stronger variant of CRT-ACD problems, where the size of p_1 is larger than other p_i's and they take r_1 from uniform distribution over $\mathbb{Z}_{p_1}$. In that case, it can be reduced to the original ACD problem.

For proper parameters, the CRT-ACD problems are regarded to be hard. In this section, however, we show that when the auxiliary input $\mathsf{CRT}_{(p_i)}(x_0/p_i)$ is given, the CRT-ACD is solved in polynomial-time of n, η, ε. Now we define a variant of CRT-ACD, as CRT-ACD problem with auxiliary input.

Definition 4.2.2. (CRT-ACDwAI) Let $n, \eta, \varepsilon \in \mathbb{N}$, and χ_ε be a distribution over $\mathbb{Z} \cap (-2^\varepsilon, 2^\varepsilon)$. For given η bit primes $p_1, \cdots, p_n$, define $x_0 = \prod_{i=1}^{n} p_i$ and $\hat{p}_i = x_0/p_i$, for $1 \leqslant i \leqslant n$. The sampleable CRT-ACD distribution $\mathcal{D}_{\chi_\varepsilon,\eta,n}(p_1, \cdots, p_n)$ is defined as

$$\mathcal{D}_{\chi_\varepsilon,\eta,n}(p_1, \cdots, p_n) = \{\mathsf{CRT}_{(p_i)}(r_i) \mid r_i \leftarrow \chi_\varepsilon\}.$$

The CRT-ACDwAI is: For given many samples from $\mathcal{D}_{\chi_\varepsilon,\eta,n}(p_1, \cdots, p_n)$, x_0 and $\hat{P} = \mathsf{CRT}_{(p_i)}(\hat{p}_i)$, to find p_i for all i.

The auxiliary input $\hat{P}$ has a special feature which can be written as a summation of its CRT components in $\mathbb{Z}_{x_0}$. A key observation is that the equation holds over the integers when $\log n + 1 < \eta$. Using this property, we obtain a following lemma.

Lemma 4.2.2. *For given* $\hat{P} = \mathsf{CRT}_{(p_i)}(\hat{p}_i)$, $x_0 = \prod_{i=1}^{n} p_i$, *and* $a = \mathsf{CRT}_{(p_i)}(r_i) \leftarrow \mathcal{D}_{\chi_\varepsilon,\eta,n}(p_1, \cdots, p_n)$, *it satisfies:*

$$a \cdot \hat{P} \bmod x_0 = \mathsf{CRT}_{(p_i)}(r_i \cdot \hat{p}_i) = \sum_{i=1}^{n} r_i \cdot \hat{p}_i$$

if $\varepsilon + \log n + 1 < \eta$.

Proof. The first equality is correct by the definition of Chinese remainder theorem. To show that the second equality is correct, we consider the equation modulo p_i for each i. Then the left hand side is $r_i \cdot \hat{p}_i$ and the right hand side is also $r_i \cdot \hat{p}_i$, because $\hat{p}_j = 0 \mod p_i$, for $j \neq i$. Finally, the size of $\sum_{i=1}^{n} r_i \cdot \hat{p}_i$ is smaller than $n \cdot 2^{\varepsilon} \cdot 2^{(n-1)\cdot\eta}$ which is less than $x_0/2$. Hence, by the uniqueness of CRT, the second equality holds. □

This lemma transforms the modulus equation to an integer equation of $r_1, \cdots, r_n$ with unknown coefficients $\hat{p}_1, \cdots, \hat{p}_n$. Our goal is to recover r_i by using the integral equation.

Now we describe full details of solving the CRT-ACDwAI.

Constructing Matrix Equations over $\mathbb{Z}$

Now we show how to compute $p_1, \cdots, p_n$ when given polynomially many samples of the CRT-ACD from $\mathcal{D}_{\chi_\varepsilon,\eta,n}(p_1, \cdots, p_n)$ with $\varepsilon + \log n + 1 < \eta$ and the auxiliary input $\hat{P} = \mathsf{CRT}_{(p_i)}(\hat{p}_i)$. For given two instances of CRT-ACD $a = \mathsf{CRT}_{(p_i)}(a_i)$ and $b = \mathsf{CRT}_{(p_i)}(b_i)$, $ab\hat{P} \bmod x_0 = \sum a_i b_i \hat{p}_i \bmod x_0$. If all of a_i's and b_i's are small enough, the right hand side equals to $\sum a_i b_i \hat{p}_i$, and so it can be written as the following matrix equation over the integers:

$$ab\hat{P} \bmod x_0 = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \end{pmatrix} \begin{pmatrix} \hat{p}_1 & 0 & \cdots & 0 \\ 0 & \hat{p}_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \hat{p}_n \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

The matrix representations share the diagonal matrix $\mathsf{diag}(\hat{p}_1, \cdots, \hat{p}_n)$ for any $\mathsf{CRT\text{-}ACD}$ instances a and b. Hence, we can construct an $(n \times n)$-matrix which is a multiple of $\mathsf{diag}(\hat{p}_1, \cdots, \hat{p}_n)$ by arranging $ab\hat{P} \mod x_0$ for various a and b.

More precisely, we are given $2n + 1$ number of samples from the distribution $\mathcal{D}_{\chi_\varepsilon,\eta,n}$ as following:

$$a_i = \mathsf{CRT}_{(p_k)}(a_{k,i}), b = \mathsf{CRT}_{(p_k)}(b_k), c_j = \mathsf{CRT}_{(p_k)}(c_{k,j}) \text{ for } 1 \leqslant i, j \leqslant n.$$

To adapt **Lemma 4.2.2** to $a_i b c_j \mod x_0$, we assume that the parameters of the problem satisfy the condition: $3\varepsilon + \log n + 1 < \eta$. Then compute the following values by multiplying the samples:

$$w_{i,j} = a_i \cdot b \cdot c_j \cdot \hat{P} \bmod x_0 = \sum_{k=1}^{n} a_{k,i} \cdot b_k \hat{p}_k \cdot c_{k,j} \text{ for } 1 \leqslant i, j \leqslant n,$$

$$w'_{i,j} = a_i \cdot c_j \cdot \hat{P} \bmod x_0 = \sum_{k=1}^{n} a_{k,i} \cdot \hat{p}_k \cdot c_{k.j} \text{ for } 1 \leqslant i, j \leqslant n.$$

They can be written as the the following matrix form:

$$w_{i,j} = \sum_{i=1}^{n} a_i \cdot \hat{p}_i b_i \cdot c_i = \begin{pmatrix} a_{1,i} & a_{2,i} & \cdots & a_{n,i} \end{pmatrix} \begin{pmatrix} b_1\hat{p}_1 & 0 & \cdots & 0 \\ 0 & b_2\hat{p}_2 & \cdots & 0 \\ 0 & 0 & \vdots & 0 \\ 0 & 0 & \cdots & b_n\hat{p}_n \end{pmatrix} \begin{pmatrix} c_{1,j} \\ c_{2,j} \\ \vdots \\ c_{n,j} \end{pmatrix}$$

$$w'_{i,j} = \sum_{i=1}^{n} a_i \cdot \hat{p}_i \cdot c_i = \begin{pmatrix} a_{1,i} & a_{2,i} & \cdots & a_{n,i} \end{pmatrix} \begin{pmatrix} \hat{p}_1 & 0 & \cdots & 0 \\ 0 & \hat{p}_2 & \cdots & 0 \\ 0 & 0 & \vdots & 0 \\ 0 & 0 & \cdots & \hat{p}_n \end{pmatrix} \begin{pmatrix} c_{1,j} \\ c_{2,j} \\ \vdots \\ c_{n,j} \end{pmatrix}$$

By collecting these values, we can construct two matrices $\mathbf{W} = (w_{i,j})$ and

$\mathbf{W}' = (w'_{i,j}) \in M_{n\times n}(\mathbb{Z})$, which can be written as

$$\mathbf{W} = \mathbf{A}^T \cdot \mathsf{diag}(b_1\hat{p_1}, \cdots, b_n\hat{p_n}) \cdot \mathbf{C},$$

$$\mathbf{W}' = \mathbf{A}^T \cdot \mathsf{diag}(\hat{p_1}, \cdots, \hat{p_n}) \cdot \mathbf{C}$$

for $\mathbf{A}^T = (a_{k,i})$ and $\mathbf{C} = (c_{k,j}) \in M_{n\times n}(\mathbb{Z})$.

Disclosing all the Secret Quantities

Suppose $\mathbf{A}$ and $\mathbf{C}$ are invertible matrices over Q. We compute $(\mathbf{W}')^{-1}$ over $\mathbb{Q}$ and the following matrix:

$$\mathbf{V} = \mathbf{W} \cdot (\mathbf{W}')^{-1} = \mathbf{A}^T \cdot \mathsf{diag}(b_1, \cdots, b_n) \cdot (\mathbf{A}^T)^{-1}.$$

Here the eigenvalues of the matrix $\mathbf{V}$ are exactly the set $B = \{b_1, \cdots, b_n\}$.

The set B can be computed in polynomial-time of η, n, and ε from $\mathbf{V}$ (e.g., by factoring the characteristic polynomial over $\mathbb{Z}$). The prime p_i is a common factor of both $(b - b_i)$ and x_0, and they have other common factor if and only if $b_j = b_i$ for some $j \in \{1, \cdots, n\}$. Hence if b_i's are distinct , we can get all secret integers $p_1, \cdots, p_n$.

$$\{\mathsf{GCD}(b - \beta, x_0) \mid \beta \in B\} = \{p_i \mid 1 \leqslant i \leqslant n\}.$$

Remark. The probability prob_1 that matrix $\mathbf{A}$ and $\mathbf{C}$ are invertible matrices depends on the distribution χ_ε. The probability prob_2 that $b_i \neq b_j$ for all $1 \leqslant i < j \leqslant n$ also depends on the distribution χ_ε. Our attack succeeds with probability of $\mathsf{prob}_1 \cdot \mathsf{prob}_2$. For example, this probability is overwhelming with respect to ε when χ_ε is uniform distribution over $(-2^\varepsilon, 2^\varepsilon)$. Since our attack consists of a matrix multiplication, computing a characteristic polynomial and finding roots of the polynomial, the overall cost is bounded by $\widetilde{\mathcal{O}}(n^{2+\omega} \cdot \eta)$, with $\omega \leqslant 2.38$. Hence, we obtain the following result:

Theorem 4.2.1. *Let U_ε be the uniform distribution over $(-2^\varepsilon, 2^\varepsilon) \cap \mathbb{Z}$. When*

$\varepsilon + \log n + 1 < \eta$ and given $O(n)$ $\mathsf{CRT\text{-}ACD}$ samples from $\mathcal{D}_{U_\varepsilon,\eta,n}(p_1, \cdots, p_n)$ with $x_0 = \prod_{i=1}^{n} p_i$, and $\hat{P} = \mathsf{CRT}_{(p_i)}(\hat{p}_i)$, one can recover every secret primes $p_1, \cdots, p_n$ in time $\widetilde{\mathcal{O}}(n^{2+\omega} \cdot \eta)$ with $\omega \leqslant 2.38$ and overwhelming probability to ε.

4.2.1 Application to CLT Schemes

In this section, we adapt the analysis of $\mathsf{CRT\text{-}ACD}$ with auxiliary input to CLT multilinear maps. The instances params and $\mathbf{p}_{zt}$ of the problem and the CLT multilinear map are quite similar. The encodings of CLT resemble the instances of the problem except the secret constant z. The zero-testing parameters $\mathbf{p}_{zt}$ also has a similar structure with $\hat{P}$ but contains coefficients with large size about p_i. However, when we restrict zero-testing to encodings of 0, it behaves similar to Lemma 4.2.2.

More precisely, let a be a top-level encoding of 0 and can be written by $a = \mathsf{CRT}_{(p_i)}(r_i g_i / z^\kappa)$. Hereafter since we use only one zero-testing parameter, without loss of generality, we denote $(\mathbf{p}_{zt})_1$ as $\mathbf{p}_{zt}$. As similar in Lemma 4.2.2,

$$\mathbf{p}_{zt} \cdot a \mod x_0 = \mathsf{CRT}_{p_i}(\hat{p}_i h_i r_i) = \sum_{i=1}^{n} \hat{p}_i h_i r_i$$

as long as the last quantity is smaller than $x_0/2$. By zero-testing conditions, it is always true for valid top level encodings of zero. Next, by replacing a by valid κ level encodings of zero $x'_j \cdot x'_1 \cdot x_k \cdot y^{k-1}$ or $x'_j \cdot x_k \cdot y^{k-1}$ for $1 \leqslant j, k \leqslant n$

in the above equation, for $1 \leqslant j, k \leqslant n$, we have:

$$\begin{aligned} w_{jk} &= x'_j \cdot x'_1 \cdot x_k \cdot y^{\kappa-1} \cdot \mathbf{p}_{zt} \bmod x_0 = \sum_{i=1}^{n} \hat{p}_i \cdot h_i \cdot x'_{ij} \cdot (r_i g_i + 1)^{\kappa-1} \cdot x'_{i1} \cdot r_{ik} \\ &= \sum_{i=1}^{n} x'_{ij} \cdot x'_{i1} \cdot h'_i \cdot r_{ik}, \text{ and} \\ w'_{jk} &= x'_j \cdot x_k \cdot y^{\kappa-1} \cdot \mathbf{p}_{zt} \bmod x_0 = \sum_{i=1}^{n} \hat{p}_i \cdot h_i \cdot x'_{ij} \cdot (r_i g_i + 1)^{\kappa-1} \cdot r_{ik} \\ &= \sum_{i=1}^{n} x'_{ij} \cdot h'_i \cdot r_{ik}, \end{aligned}$$

where $h'_i = \hat{p}_i \cdot h_i \cdot (r_i g_i + 1)^{\kappa-1}$. By spanning $1 \leqslant i, j \leqslant n$, we obtain the matrix $\mathbf{W}$ and $\mathbf{W}'$:

$$\mathbf{W} = \mathbf{X}'^T \cdot \mathsf{diag}(x'_{11} \cdot h'_1, \cdots, x'_{n1} \cdot h'_n) \cdot \mathbf{R},$$

$$\mathbf{W}' = \mathbf{X}'^T \cdot \mathsf{diag}(h'_1, \cdots, h'_n) \cdot \mathbf{R},$$

for $\mathbf{X}'^T = (x'_{ij})$ and $R = (r_{ik})$. By applying the same method in the section 2, we can recover $\{x_{11}, \cdots, x_{n1}\}$ by computing the eigenvalues of $\mathbf{W} \cdot \mathbf{W}'^{-1}$. Hence we can compute all secret p_i by computing $\mathsf{GCD}(x'_1 - x_{i1}, x_0)$.

Consequently, we need $\mathbf{W}'$ and $\mathbf{W}$ to be invertible. We argue that this is the case here. We prove it for $\mathbf{W}$. Note first that the x'_{i1}'s and the h'_i's are all non-zero, with overwhelming probability. Note that by design, the matrix $(r_{ij})_{i\in[n],j\in[\tau]}$ has rank n (see [CLT13, Section. 4]). The same holds for the matrix $(x'_{ij})_{i\in[n],j\in[\ell]}$ (see [CLT13, Lemma. 1]). As we can compute the rank of a $\mathbf{W} \in \mathbb{Z}^{t\times t}$ obtained by using an $\mathbf{X'} \in \mathbb{Z}^{t\times n}$ and an $\mathbf{R} \in \mathbb{Z}^{n\times t}$ obtained by respectively using a t-subset of the x'_j's and a t-subset of the x_j's. Without loss of generality we may assume that our $\mathbf{X}', \mathbf{R} \in \mathbb{Z}^{\mathbf{n}\times\mathbf{n}}$ are non-singular. The cost of finding such a pair $(\mathbf{X}', \mathbf{R})$ is bounded as $\widetilde{\mathcal{O}}((\tau + \ell) \cdot (n^\omega \log x_0)) = \widetilde{\mathcal{O}}(\kappa^{\omega+3}\lambda^{2\omega+6})$, with $\omega \leqslant 2.38$ (assuming all parameters are set smallest possible so that the bounds of parameter setting hold). Here we used the fact that the rank of a matrix $\mathbf{A} \in \mathbb{Z}^{n\times n}$ may be computed

in time $\widetilde{\mathcal{O}}(n^\omega \log \|\mathbf{A}\|_\infty)$ (see [Sto09]). This dominates the overall cost of the attack.

After we know all the p_i's, we have $x_j/y = r_{ij}g_i/(r_i g_i + 1) \bmod p_i$. As the numerator and denominator are coprime and very small compared to p_i, they can be recovered by the rational reconstruction algorithm. We hence obtain $(r_{ij}g_i)$'s for all j. The gcd of all the $(r_{ij}g_i)$'s reveals g_i. As a result, we can also recover all the r_{ij}'s and r_i's. As $x_1 = r_{i1}g_i/z \bmod p_i$ and the numerator is known, we can recover $z \bmod p_i$ for all i, and hence $z \bmod x_0$. The h_{ij}'s can then be recovered as well, so can the r'_{ij}'s and a_{ij}'s.

Related and Follow-up Works. Our attack was extended in [BWZ14, GHMS14, CGH$^+$15] to settings in which no low-level encoding of 0 are available. The extensions rely on low-level encodings of elements corresponding to orthogonal vectors and impact [GLW14, GLSW15, GGH$^+$13b].

After our attack was published in Eurocrypt'15, the draft [GGHZ14] was update to propose a candidate immunization against our attack (see [GGHZ14, Se. 6]).[‡] Another candidate immunization was proposed in [BWZ14]. Both immunizations have proved insecure in [CLT14a]. See also [CGH$^+$15].

A further modification of CLT was proposed by Coron, Lepoint and Tibouchi in the proceedings of CRYPTO'15 [CLT15]. They claimed that our attack is thwarted since the modified scheme keeps the modulus secret so that the zero-testing procedure depends on the CRT components in a non-linear way. However, it turned out to be insecure as proved by Cheon *et al.* in [CFL$^+$16] who exploit an extension of eigenvalues and determinant techniques as in **Section 4.2** and **4.3**.

Remark. In case of the obfuscation on CLT multilinear map, the security remained open problems because the applications is not given an encodings of zero. Recently, Coron *et al.* provide a new result [CLLT16] about it, which enables one to break the obfuscation on CLT multilinear map in polynomial-

[‡]The former version that was impacted by our attack can still be accessed from the IACR eprint server.

time.

4.3 Analysis of the Related Problems.

In this section, we provide an analysis of the SubM, DLIN, and GXDH problems associated with the CLT multilinear map. We start by defining these problems adapted to CLT version. We then describe how to solve these problems in polynomial-time. Briefly,these problems are unified by the problem of determining whether a given matrix is full rank or not. The only difference is the dimension of a given matrix. SubM is given rank n, L-DLIN is Ln, and GXDH is $2n$. Therefore, the attack algorithm of these problems is the same process.

The attack procedure consists of two steps. First, in Section 4.3, we show how to recover $\prod_i g_i$. Next, in Sections 4.3.1 and 4.3.2, we use that quantity to recognize valid instances of the SubM and DLIN. In Section 4.3.3, we introduce a method to solve the GXDH.

Let $G = \mathbb{Z}_{g_1} \times \ldots \times \mathbb{Z}_{g_n}$ and G_i be the subgroup of order g_i obtained by making the components of the other $\mathbb{Z}_{g_j}$'s to be zero. For index set $I \subseteq [n]$, we denote $G_I = \prod_{i \in I} G_i$. We let $\mathsf{enc}_1(m)$ denote a properly generated level-1 encoding of $m \in G$. For integers $L, N > 0$, we let $\mathsf{Rk}_i(\mathbb{Z}_N^{L \times L})$ denote the set of $L \times L$ matrices over $\mathbb{Z}_N$ of rank i. If N is a product of primes, we define the rank of a matrix as the maximum of the ranks of the matrices obtained by reduction modulo all the prime divisors of N.

Definition 4.3.1. (The Subgroup Membership Problem) SubM is as follows. Given λ and κ, generate params and $\mathbf{p_{zt}}$ using $\mathsf{InstGen}$ and $\{\mathsf{enc}_1(g_i) : i \in [\ell]\}$ where the g_i's are uniformly and independently sampled in a strict subgroup G_I of G, with ℓ sufficiently large so that the g_i's generate G_I with overwhelming probability. Given params, $\mathbf{p}_{zt}$, $\{\mathsf{enc}_1(g_i) : i \in [\ell]\}$ and $u = \mathsf{enc}_1(m)$, determine whether m is sampled uniformly in G_I or in G.

Definition 4.3.2. (L-Decisional Linear Problem) L-DLIN is as follows. Given λ and κ, generate params and $\mathbf{p}_{zt}$ using $\mathsf{InstGen}$. Define $N = \prod_i g_i$.

Given params and $\mathbf{p}_{zt}$, the goal is to distinguish between the distributions

$$\{(\mathsf{enc}_1(m^{(i,j)}))_{i,j}\}_{(m^{(i,j)})_{i,j}\leftarrow\mathsf{Rk}_{L-1}(\mathbb{Z}_N^{L\times L})} \text{ and } \{(\mathsf{enc}_1(\tilde{m}^{(i,j)}))_{i,j}\}_{(\tilde{m}^{(i,j)})_{i,j}\leftarrow\mathsf{Rk}_{L}(\mathbb{Z}_N^{L\times L})}.$$

In one of the constructions of [ABP15], the authors rely on the following particular case. The problem is as follows. The algorithm is given params and $\mathbf{p}_{zt}$ as well as $\{\mathsf{enc}_1(a_i)\}_{i\in[L]}$ and $\{\mathsf{enc}_1(a_ib_i)\}_{i\in[L]}$ for some uniform and independent $a_1,\ldots,a_L,b_1,\ldots,b_L\in G$. It is also given $\mathsf{enc}_1(m)$, and it has to assess whether m is uniformly and independently sampled in G or whether $m = b_1+\ldots+b_L$. This can be restated as a special case of Definition 4.3.2, by noting that it requests to assess whether the matrix just below is full-rank.

$$\begin{pmatrix} a_1b_1 & a_1 & 0 & \ldots & 0 \\ a_2b_2 & 0 & a_2 & \ldots & 0 \\ & \vdots & & & \\ a_Lb_L & 0 & 0 & \ldots & a_L \\ m & 1 & 1 & \ldots & 1 \end{pmatrix}$$

We recall asymmetric multilinear maps and the associated GXDH problem. By applying the attacks described above, we can solve GXDH in polynomial-time.

Instance generation: $(\mathsf{params},\mathbf{p}_{zt})\leftarrow\mathsf{InstGen}(1^\lambda,1^\kappa)$. The setting of the parameters $p_i, g_i,\ x_0, \{x'_j\}, \Pi$ and H are as in the original scheme. For $1\leqslant t\leqslant\kappa$, sample z_t uniformly in $\mathbb{Z}_{x_0}$. Then define, for all $1\leqslant t\leqslant\kappa$:

$$\begin{aligned} y^{(t)} &= \mathsf{CRT}_{(p_i)}\left(\frac{r_i^{(t)}\cdot g_i+1}{z_t}\right),\ \text{where } r_i^{(t)}\leftarrow(-2^\rho,2^\rho)\cap\mathbb{Z},\ \text{for } 1\leqslant i\leqslant n, \\ x_j^{(t)} &= \mathsf{CRT}_{(p_i)}\left(\frac{r_{ij}^{(t)}\cdot g_i}{z_t}\right),\ \text{for } 1\leqslant j\leqslant\tau. \end{aligned}$$

Further, we define:

$$(\mathbf{p}_{zt})_j = \sum_{i=1}^{n} h_{ij} \cdot \Big(\prod_{1 \leqslant t \leqslant \kappa} z_t \cdot g_i^{-1} \bmod p_i \Big) \cdot \prod_{i' \neq i} p_{i'} \bmod x_0, \text{ for } 1 \leqslant j \leqslant n.$$

Output $\mathsf{params} = (n, \eta, \alpha, \rho, \beta, \tau, \ell, \nu, x_0\{y^{(t)}\}, \{x_j^{(t)}\}, \{x'_j\}, \{\Pi_j\}, s)$ and $\mathbf{p}_{zt}$. From now on, we let $\mathsf{enc}_t(m)$ denote $\mathsf{CRT}_{(p_i)}(\frac{m_i + s_i \cdot g_i}{z_t})$.

Since, as same as in section 3.2, we only use one zero-testing parameter, we denote $(\mathbf{p}_{zt})_1$ as $\mathbf{p}_{zt}$. We now define the CLT variant of the GXDH problem.

Definition 4.3.3. (Graded External DDH Problem) GXDH is as follows. Given λ and κ, generate params and $\mathbf{p}_{zt}$ using $\mathsf{InstGen}$. Given params, $\mathbf{p}_{zt}$ and $\mathsf{enc}_t(a), \mathsf{enc}_t(b)$ and $\mathsf{enc}_t(c)$ with $a, b \leftarrow G$ and for a given $t \in [\kappa]$, the goal is to decide whether $c = a \cdot b$ or c is uniformly and independently sampled in G.

This can be regarded as a variant of 2-DLIN problems by distinguishing the following distributions

$$\left\{ \begin{pmatrix} \mathsf{enc}_t(c) & \mathsf{enc}_t(a) \\ \mathsf{enc}_t(b) & \mathsf{enc}_t(1) \end{pmatrix} \right\} \text{ and } \left\{ \begin{pmatrix} \mathsf{enc}_t(ab) & \mathsf{enc}_t(a) \\ \mathsf{enc}_t(b) & \mathsf{enc}_t(1) \end{pmatrix} \right\}, \text{where } c \leftarrow G.$$

Our main strategy to solve these three related problem of CLT scheme is that: For a given level-1 encoding

$$\mathbf{E} = (e_{i,j}) = \mathsf{CRT}_{(p_k)} \left(\frac{s_k^{(i,j)} g_k + m_k^{(i,j)}}{z} \right) \text{ for } 1 \leqslant i, j \leqslant t,$$

we can construct a matrix $\mathbf{W}_{e_{i,j}} := \mathbf{W}_{i,j}$ as similar to **Section 4.2** by computing $[x'_k \cdot e_{i,j} \cdot x_l \cdot y^{\kappa-2} \cdot \mathbf{p}_{zt}]_{x_0}$ for $1 \leqslant k, l \leqslant n$:

$$\begin{aligned} \mathbf{W}_{i,j} &= \mathbf{X}' \cdot (\mathbf{S}_{i,j}\mathbf{G} + \mathbf{M}_{i,j}) \cdot \mathsf{diag}(\tilde{h}_1, \cdots, \tilde{h}_n) \cdot \mathbf{R} \\ &= \mathbf{X}' \cdot (\mathbf{S}_{i,j}\mathbf{G} + \mathbf{M}_{i,j}) \cdot \mathbf{R}', \end{aligned}$$

where $\tilde{h}_i = h_i \cdot (r_i g_i + 1)^{\kappa-2} \cdot \hat{p}_i$, $\mathbf{S}_{i,j} = \mathsf{diag}(s_1^{(i,j)}, \cdots, s_n^{(i,j)})$ and $\mathbf{M}_{i,j} =$

$\mathsf{diag}(m_1^{(i,j)}, \cdots, m_n^{(i,j)})$. By collecting these matrix $\mathbf{W} = (\mathbf{W}_{i,j})$ for $1 \leqslant i, j \leqslant t$, we can get following matrix:

$$\mathbf{W} = \mathbf{X}' \cdot \left(\begin{pmatrix} \mathbf{S}_{1,1} \cdot \mathbf{G} & \dots & \mathbf{S}_{1,t} \cdot \mathbf{G} \\ \vdots & \ddots & \vdots \\ \mathbf{S}_{t,1} \cdot \mathbf{G} & \dots & \mathbf{S}_{t,t} \cdot \mathbf{G} \end{pmatrix} + \begin{pmatrix} \mathbf{M}_{1,1} & \dots & \mathbf{M}_{1,t} \\ \vdots & \ddots & \vdots \\ \mathbf{M}_{t,1} & \dots & \mathbf{M}_{t,t} \end{pmatrix} \right) \cdot \mathbf{R}'.$$

Related problems are to distinguish problems for given matrix of encoding $\mathbf{E}$, the size of matrix is different depending on problem. Those related problems can be seen as following:

SubM: For $t = 1$ and a given $\mathbf{E}$, determine $m \leftarrow G_I$ or not.

L-DLIN: For $t = L$ and a given $\mathbf{E}$, determine $(m^{(i,j)})_{i,j} \leftarrow \mathsf{Rk}_{L-1}(\mathbb{Z}_N^{L \times L})$ or $\mathsf{Rk}_L(\mathbb{Z}_N^{L \times L})$.

GXDH: For $t = 2$ and a given $\mathbf{E}$, determine $\begin{pmatrix} c & a \\ b & 1 \end{pmatrix}$ is a full rank or not

In case of SubM, determining $\mathbf{m} = (m_i)_{1 \leqslant i \leqslant n}$ is in G_I or not is the same as computing factors of $\mathsf{gcd}(\prod(r_i g_i + m_i), \prod g_i)$. This value can be computed from determinant of $\mathbf{W}$ and $\prod g_i$. In case of GXDH and L-DLIN, the determinant of $\mathbf{W}$ is a multiple of g_i for any i, if the middle term matrix $\mathbf{M}$ does not have a full rank. In other case, the determinant of $\mathbf{M}$ is not a multiple of g_i with a high probability. Hence, if one can recover the $\prod g_i$, one can solve the related problems.

Remark. The important difference between cryptanalysis of these related problems and the cryptanalysis of the CLT scheme is the form of the middle matrix of $\mathbf{W}$. The previous attack in Section 3 is based on the fact that the middle matrix is a diagonal matrix. For example, in [BWZ14], the authors fixed the middle matrix into block diagonal matrix form.[§] On the other hand, the attack of related problems in this section does not depend on it.

[§]Soon after, it is also known to be insecure by Coron *et al.*'s extended attack

Step 1: Computing $\prod_i g_i$

Suppose we have public instances

$$\mathsf{params} = (n, \eta, \alpha, \rho, \beta, \tau, \ell, \nu, x_0, y, \{x_j\}, \{x'_j\}, \{\Pi_j\}, s) \text{ and } \mathbf{p}_{zt}.$$

The main step in the attack is to get $\prod_i g_i$ from $(\mathsf{params}, \mathbf{p}_{zt})$. It may be admissible to assume that the g_i's are public in which computing $\prod_i g_i$ is trivial. If for some reason the g_i's have to stay secret, one must set their bit-sizes as $\Omega(\lambda^2)$, so that they cannot be recovered by combining the approach described below with the elliptic curve factorization algorithm.

Similarly to the **Section 4.2**, we compute $w_{kl} := [x'_k \cdot y \cdot x_l \cdot y^{\kappa-2} \cdot \mathbf{p}_{zt}]_{x_0}$, $w^{(i)}_{kl} := [x'_k \cdot x_i \cdot x_l \cdot y^{\kappa-2} \cdot \mathbf{p}_{zt}]_{x_0}$ and obtain a matrix

$$\begin{aligned} \mathbf{W}_y &= \mathbf{X}' \cdot \mathsf{diag}(r_1 g_1 + 1, \ldots, r_n g_n + 1) \cdot \mathbf{R}'. \\ \mathbf{W}_i &= \mathbf{X}' \cdot \mathsf{diag}(r_{i1} g_1, \ldots, r_{in} g_n) \cdot \mathbf{R}'. \end{aligned}$$

We can get a multiple of $\prod_i g_i$ by taking a ratio of gcd's of determinants of appropriate subsets of $\{\mathbf{W}_1, \ldots, \mathbf{W}_m, \mathbf{W}_y\}$:

$$\begin{aligned} & \frac{\gcd(\det \mathbf{W}_1, \ldots, \det \mathbf{W}_m)}{\gcd(\det \mathbf{W}_1, \ldots, \det \mathbf{W}_m, \det \mathbf{W}_y)} \\ = \; & \frac{\gcd(\prod_i r_{i1}, \ldots, \prod_i r_{im})}{\gcd(\prod_i r_{i1} g_i, \ldots, \prod_i r_{im} g_i, \prod_i (r_i g_i + 1))} \cdot \prod_i g_i \\ = \; & \Delta \cdot \prod_i g_i, \end{aligned}$$

for some integer Δ. We expect that Δ consists of only small factors because it is a common divisor of many random variables. These variables do not satisfy uniformity condition, because r_{ij} is chosen in a half-open parallelepiped spanned by matrix Π. However the elements of matrix Π are drawn from some interval that is independent of an arbitrary prime p. Therefore, we may (heuristically) assume that the smoothness probabilities are the same as that of the uniform case. Under this assumption, the integer Δ is $2n$-smooth (i.e.,

all its divisors are $\leqslant 2n$) with probability $\geqslant 0.9$, as we explain below. The more general results can be found in [CK16].

Lemma 4.3.3 (Heuristic)**.** *Let r_{ij} be a random integer for $i \in [n], j \in [m]$ with $m \geqslant s \log(2n)$ for some positive integer s. Then $\gcd(\prod_i r_{i1}, \ldots, \prod_i r_{im})$ is $2n$-smooth with probability $\geqslant \zeta(s)^{-1}$, which is $\geqslant 0.9$ when $s \geqslant 4$.*

Proof. Our heuristic assumption is that each r_{ij} is divisible by a prime $p > 2n$ with probability $\leqslant 1/p$, for all p's. First, we observe that for each j, the integer $\prod_i r_{ij}$ is divisible by p with probability $\leqslant 1 - (1 - 1/p)^n \leqslant n/p$. Then the probability that $\gcd(\prod_i r_{i1}, \ldots, \prod_i r_{im})$ is divisible by p is $\leqslant (n/p)^m$. As a result, the gcd is $2n$-smooth with probability at least

$$\prod_{p>2n} (1 - (n/p)^m) \geqslant \prod_{p>2n} (1 - 1/p^s) = \zeta(s)^{-1} \prod_{p \leqslant 2n} (1 - 1/p^s)^{-1} \geqslant \zeta(s)^{-1}.$$

Here the first inequality comes from $(n/p)^m \leqslant (n/2n)^m = (1/2)^m \leqslant 1/p^s$ for $m \geqslant s \log p$. The equality is Euler's identity for the Riemann zeta function. The latter is decreasing and $\zeta(4)^{-1} > 0.9$. This completes the proof. □

By Lemma 4.3.3, the integer Δ is $(2n)$-smooth with probability > 0.9. We eliminate it by trial division by all integers $\leqslant 2n$. This costs $\widetilde{\mathcal{O}}(\kappa^2 \lambda^5)$ bit operations. This is dominated by the cost of the operations described in Sections 4.2, which is $\widetilde{\mathcal{O}}(\kappa^{\omega+3} \lambda^{2\omega+6})$.

4.3.1 Solving the CLT SubM Problem

We compute $w_{kl} = [x'_k \cdot \mathsf{enc}_1(m) \cdot x_l \cdot y^{\kappa-2} \cdot \mathbf{p}_{zt}]_{x_0}$:

$$\mathbf{W} = \mathbf{X}' \cdot \mathsf{diag}(r_1 g_1 + x_1, \ldots, r_n g_n + x_n) \cdot \mathbf{R}',$$

with $x_i \in \mathbb{Z}_{g_i}$ for all i. The attack consists in computing $\gcd(\det \mathbf{W}, \prod_i g_i)$.

If m is uniformly sampled in G, then we expect $n/2^\alpha$ of the x_i's to be zero. Hence, in that case, we have $\log \gcd(\det \mathbf{W}, \prod_i g_i) \approx \alpha n / 2^\alpha$. For the original setting of $\alpha = \lambda$, this is essentially 0.

If m is uniformly sampled in G_I, then all the x_i's for $i \notin I$ are zero, and we expect $(n - |I|)/2^\alpha$ of the others to be zero. Hence, in that case, we have $\log \gcd(\det \mathbf{W}, \prod_i g_i) \approx \alpha|I| + \alpha(n - |I|)/2^\alpha$.

4.3.2 Solving the CLT DLIN Problem

As we have seen, we assume that $\prod_i g_i$ is known. In DLIN, we are given a matrix of level-1 encodings $\mathbf{E} = (e_{i,j})_{i,j}$. We write $e_{i,j} = (s_k^{(i,j)} g_k + m_k^{(i,j)})/z \bmod p_k$. Using the same method to above, we compute matrices $\mathbf{W}_{i,j} \in \mathbb{Z}^{n \times n}$ for all $e_{i,j}$. We define

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} & \dots & \mathbf{W}_{1L} \\ \mathbf{W}_{21} & \mathbf{W}_{22} & \dots & \mathbf{W}_{2L} \\ \vdots & & \ddots & \\ \mathbf{W}_{L1} & \mathbf{W}_{L2} & \dots & \mathbf{W}_{LL} \end{pmatrix} \in \mathbb{Z}^{nL \times nL}.$$

We compute the determinant of $\mathbf{W}$. It satisfies the following equation.

$$\det(\mathbf{W}) = \det(\mathbf{X}')^L \cdot \det(\mathbf{R}')^L \cdot \det \begin{pmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} & \dots & \mathbf{B}_{1,L} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \dots & \mathbf{B}_{2,L} \\ \vdots & & \ddots & \\ \mathbf{B}_{L,1} & \mathbf{B}_{L,2} & \dots & \mathbf{B}_{L,L} \end{pmatrix},$$

where $\mathbf{B}_{i,j} = \mathsf{diag}(s_1^{(i,j)} \cdot g_1 + m_1^{(i,j)}, \cdots, s_n^{(i,j)} \cdot g_n + m_n^{(i,j)})$ for all i, j. Let $\Delta = \det(\mathbf{X}')^L \cdot \det(\mathbf{R}')^L$. We have $\det \mathbf{W} = \Delta \cdot \prod_k \det \mathbf{Q}_k$, where $\mathbf{Q}_k = (r_k^{(i,j)} \cdot g_k + m_k^{(i,j)})_{i,j}$ and it is congruent to $\mathbf{P}_k = (m_k^{(i,j)})_{(i,j)}$ in modulo g_k.

To distinguish among the instances of DLIN, we compute $\det \mathbf{W}$ and check whether it is divisible by $\prod_k g_k$. If $\mathbf{E}$ is sampled from a full rank matrix, the determinant of $\mathbf{P}_k$ is nonzero for some k. Hence $\det \mathbf{W}$ cannot be multiple of $\prod_k g_k$. In other case, then $\det \mathbf{P}_i = 0$ for all i. Hence $\det \mathbf{W}$ is a multiple of $\prod_k g_k$. The total bit-complexity of the attack is $\widetilde{\mathcal{O}}(\kappa^{\omega+3}\lambda^{2\omega+6} + \kappa^{\omega+3}L^{\omega+1}\lambda^{2\omega+5})$.

4.3.3 Solving the CLT GXDH Problem

In the following, we assume that $\kappa \geqslant 3$. Without loss of generality, we assume that $t = 1$ in the GXDH problem. The first step in the attack is to get $\prod_i g_i$ from $(\mathsf{params}, \mathbf{p}_{zt})$. Similar to Section 4.1, we compute $\mathbf{W}_{y^{(1)}}$ and the $\mathbf{W}_i$'s by using (params), as follows (for $1 \leqslant i \leqslant m$):

$$\begin{aligned}
\mathbf{W}_{y^{(1)}} &= ([y^{(1)} \cdot x_k^{(2)} x_l^{(3)} \cdot y^{(4)} \dots y^{(\kappa)} \cdot \mathbf{p}_{zt}]_{x_0})_{k,l} \\
&= \mathbf{R} \cdot \mathsf{diag}(r_1^{(1)} g_1 + 1, \dots, r_n^{(1)} g_n + 1) \cdot \mathsf{diag}(h'_1, \dots, h'_n) \cdot \mathbf{R}', \\
\mathbf{W}_i &= ([x_i^{(1)} \cdot x_k^{(2)} x_l^{(3)} \cdot y^{(4)} \dots y^{(\kappa)} \cdot \mathbf{p}_{zt}]_{x_0})_{k,l} \\
&= \mathbf{R} \cdot \mathsf{diag}(r_{i1}^{(1)} g_1, \dots, r_{in}^{(1)} g_n) \cdot \mathsf{diag}(h'_1, \dots, h'_n) \cdot \mathbf{R}',
\end{aligned}$$

where $\mathbf{R} = (r_{ki}^{(2)})$ and $\mathbf{R}' = (r_{il}^{(3)})$.

Similar to Section 4.1, we obtain a multiple of $\prod_i g_i$ by taking a ratio of gcd's of determinants of appropriate subsets of $\{\mathbf{W}_1, \dots, \mathbf{W}_m, \mathbf{W}_{y^{(1)}}\}$:

$$\frac{\gcd(\det \mathbf{W}_1, \dots, \det \mathbf{W}_m)}{\gcd(\det \mathbf{W}_1, \dots, \det \mathbf{W}_m, \det \mathbf{W}_{y^{(1)}})} = \Delta \cdot \prod_i g_i,$$

for some integer Δ. For the same reason as before, by Lemma 4.3.3, the integer Δ is $(2n)$-smooth with probability > 0.9. We eliminate it by trial division by all integers $\leqslant 2n$. Thus, we can get $\prod_i g_i$ in time $\widetilde{\mathcal{O}}(\kappa^{\omega+3}\lambda^{2\omega+6})$.

Next, we instantiate with $y^{(1)} = \mathsf{enc}_1(a), \mathsf{enc}_1(b), \mathsf{enc}_1(c)$, respectively. We get:

$$\begin{aligned}
\mathbf{W}_a &= \mathbf{R} \cdot \mathsf{diag}(r_{a1}^{(1)} g_1 + a_1, \dots, r_{an}^{(1)} g_n + a_n) \cdot \mathsf{diag}(h'_1, \dots, h'_n) \cdot \mathbf{R}', \\
\mathbf{W}_b &= \mathbf{R} \cdot \mathsf{diag}(r_{b1}^{(1)} g_1 + b_1, \dots, r_{bn}^{(1)} g_n + b_n) \cdot \mathsf{diag}(h'_1, \dots, h'_n) \cdot \mathbf{R}', \\
\mathbf{W}_c &= \mathbf{R} \cdot \mathsf{diag}(r_{c1}^{(1)} g_1 + c_1, \dots, r_{cn}^{(1)} g_n + c_n) \cdot \mathsf{diag}(h'_1, \dots, h'_n) \cdot \mathbf{R}'.
\end{aligned}$$

Then, we can compute:

$$\begin{aligned}
\mathbf{W} &= \begin{pmatrix} \mathbf{W}_c & \mathbf{W}_a \\ \mathbf{W}_b & \mathbf{W}_{y^{(1)}} \end{pmatrix} \in \mathbb{Z}^{2n \times 2n} \text{ and} \\
\det \mathbf{W} &= \Delta' \cdot \left((r_{ai}^{(1)} g_i + a_i) \cdot (r_{bi}^{(1)} g_i + b_i) - (r_{ci}^{(1)} g_i + c_i) \cdot (r_i^{(1)} g_i + 1) \right),
\end{aligned}$$

where $\Delta' = \det(R)^2 \cdot \det(R')^2 \cdot (\prod_i h'_i)^2$. If c is equal to $a \cdot b$, then the quantity above has $\prod_i g_i$ as a large factor. If c is uniformly and independently sampled in G, then the quantity above is independent from $\prod_i g_i$. The cost of the attack is bounded by $\widetilde{\mathcal{O}}(\kappa^{\omega+3}\lambda^{2\omega+6})$.

Chapter 5

Conclusions

In this thesis, we present the analysis of two candidate of multilinear maps of CLT and GGH.

First, in case of GGH, we have shown that if the low level encodings of zero exist, the underlying problem is reduced to the SVP on ideal lattice. By using the sublattice algorithm, we also show that given parameters in GGH is insecure.

If we have only a top level encoding of zero, we reduce from the underlying problem GDDH of the GGH to the ONTRU problem. In this work, we described how to find a small solution of the variant of the NTRU problem using a reduction technique. By applying our proposed algorithm, subfield algorithm, to the GGH scheme, we could attack the GDDH problem in the GGH scheme. Therefore, our results imply that there is no guarantee for the security of the GGH scheme when we are given a small encoding of zero and also when we are not given.

Unfortunately, the current attacks of the GGH scheme can be resist by increasing the size of dimension n from λ^2 to λ^3, where λ is a security parameter. Thus, the natural extension of this research is to increase the range of possible attack parameters.

Finally, we propose polynomial-time attacks for CRT-ACD with auxiliary input, the CLT scheme and its related problems. Until now, the CRT-ACD

is known to be hard problems. However, if an auxiliary input $\hat{P} = \sum_{i=1}^{n} \prod_{j \neq i} p_j$ is given, we find quadratic equations for secret parameters and construct a matrix. The matrix has eigenvalues as secret parameters and reveals them by computing characteristic polynomial of the matrix. Adapting this methods to the CLT scheme allows us to totally find every secret parameters.

Unfortunately, this analysis is possible only when low level encodings of zero and zero-testing parameter are given. To date, there is no known how to analyze the CLT scheme when there is no encoding of zero. Therefore, natural proceedings of this research is to extend the range of applications of graded encoding schemes for which the encodings of zero are not needed.

Bibliography

[ABD16] Martin Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched ntru assumptions. In *Annual Cryptology Conference*, pages 153–178. Springer, 2016.

[ABP15] Michel Abdalla, Fabrice Benhamouda, and David Pointcheval. Disjunctions for hash proof systems: New constructions and applications. In *Advances in Cryptology - EUROCRYPT 2015*, pages 69–100, 2015.

[ADRSD14] Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the shortest vector problem in 2^n time via discrete gaussian sampling. *arXiv preprint arXiv:1412.7994*, 2014.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology - CRYPTO 2004*, pages 41–55, 2004.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005*, pages 325–341, 2005.

[BLMR13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic prfs and their ap-

plications. In *Advances in Cryptology - CRYPTO 2013*, pages 410–428, 2013.

[BS03] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics, American Mathematical Society*, 324:71–90, 2003.

[BWZ14] Dan Boneh, David J. Wu, and Joe Zimmerman. Immunizing multilinear maps against zeroizing attacks. *IACR Cryptology ePrint Archive*, 2014.

[CCK+13] Jung Hee Cheon, Jean-Sébastien Coron, Jinsu Kim, Moon Sung Lee, Tancrede Lepoint, Mehdi Tibouchi, and Aaram Yun. Batch fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 315–335. Springer, 2013.

[CFL+16] Jung Hee Cheon, Pierre-Alain Fouque, Changmin Lee, Brice Minaud, and Hansol Ryu. Cryptanalysis of the new CLT multilinear map over the integers. *IACR Cryptology ePrint Archive*, 2016.

[CGH+15] Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K. Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In *Advances in Cryptology - CRYPTO 2015*, pages 247–266, 2015.

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *Advances in Cryptology - EUROCRYPT 2015*, pages 3–12, 2015.

BIBLIOGRAPHY

[CJL16] Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. An algorithm for ntru problems and cryptanalysis of the ggh multilinear map without a low level encoding of zero. *Mh*, 1:0, 2016.

[CK16] Jung Hee Cheon and Duhyeong Kim. Probability that the k-gcd of products of positive integers is b-smooth. *IACR Cryptology ePrint Archive*, page 334, 2016.

[CL15] Jung Hee Cheon and Changmin Lee. Approximate algorithms on lattices with small determinant. Technical report, Cryptology ePrint Archive, Report 2015/461, 2015.

[CLLT15] Jean-Sebastien Coron, Moon Sung Lee, Tancrede Lepoint, and Mehdi Tibouchi. Cryptanalysis of ggh15 multilinear maps. 2015.

[CLLT16] Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Zeroizing attacks on indistinguishability obfuscation over clt13. 2016.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *Advances in Cryptology - CRYPTO 2013*, pages 476–493, 2013.

[CLT14a] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Cryptanalysis of two candidate fixes of multilinear maps over the integers. *IACR Cryptology ePrint Archive*, 2014.

[CLT14b] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Personal communication. 2014.

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In *Advances in Cryptology - CRYPTO 2015*, pages 267–286, 2015.

BIBLIOGRAPHY

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *Advances in Cryptology - EUROCRYPT 2013*, pages 1–17, 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *IEEE Symposium on Foundations of Computer Science, FOCS*, pages 40–49, 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices, 2015.

[GGHZ14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Fully secure attribute based encryption from multilinear maps. *IACR Cryptology ePrint Archive*, 2014.

[GHMS14] Craig Gentry, Shai Halevi, Hemanta K. Maji, and Amit Sahai. Zeroizing without zeroes: Cryptanalyzing multilinear maps without encodings of zero. *IACR Cryptology ePrint Archive*, 2014.

[GLSW15] Craig Gentry, Allison B. Lewko, Amit Sahai, and Brent Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. In *Proceedings of FOCS 2015*, pages 151–170, 2015.

[GLW14] Craig Gentry, Allison B. Lewko, and Brent Waters. Witness encryption from instance independent assumptions. In *Advances in Cryptology - CRYPTO 2014*, pages 426–443, 2014.

[HG01] Nick Howgrave-Graham. Approximate integer common divisors. pages 51–66, 2001.

[HJ16] Yupu Hu and Huiwen Jia. Cryptanalysis of GGH map. In *Eurocrypt*, pages 537–565, 2016.

BIBLIOGRAPHY

[HPS11] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. 2011:447–464, 2011.

[KF17] Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on overstretched ntru parameters. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–26. Springer, 2017.

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[LS14] Hyung Tae Lee and Jae Hong Seo. Security analysis of multilinear maps over the integers. In *Advances in Cryptology - CRYPTO 2014*, pages 224–240, 2014.

[MR09] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

[MW01] Daniele Micciancio and Bogdan Warinschi. A linear space algorithm for computing the hermite normal form. In *Proceedings of the 2001 international symposium on Symbolic and algebraic computation*, pages 231–236. ACM, 2001.

[Sco02] Mike Scott. Authenticated ID-based key exchange and remote log-in with simple token and PIN number. *IACR Cryptology ePrint Archive*, page 164, 2002.

[Sto09] Arne Storjohann. Integer matrix rank certification. In *Symbolic and Algebraic Computation, International Symposium, ISSAC*, pages 333–340, 2009.

[vDGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers.

In *Advances in Cryptology - EUROCRYPT 2010*, pages 24–43, 2010.

[Zim15] Joe Zimmerman. How to obfuscate programs directly. In *Advances in Cryptology - EUROCRYPT 2015*, pages 439–467, 2015.

국문초록

다중선형함수는 암호학에서 강력한 도구 중 하나이다. 그럼에도 불구하고, 현재까지 3종류의 다중선형함수만이 제안되었다. Garg 등에 의해 첫 번째로 제안된 다중선형함수(GGH)는 아이디얼 격자를 이용하여 제안되었고, Coron 등은 정수기반 다중선형함수(CLT)를 제안했다. 마지막으로 Gentry 등에 의해 그래프 기반 다중선형함수가 제안되었다. 다중선형함수의 등장은 여러 암호학적 응용들을 현실화 시키는 계기가 되었다.

이러한 응용들의 안전성은 주어진 다중선형함수로부터 파생된 어려운 문제들에 기반을 두어 설명이 되는데 아직까지 파생된 문제들의 안전성은 설명되지 못하고 있는 상황이다. 이로 인해 다중선형함수의 안전성을 설명하기 위한 후속 연구가 꾸준히 진행되고 있는데, 실제로 GGH 다중선형함수에서 낮은 등급의 0의 인코딩이 주어지는 경우와 세 번째 다중선형함수는 안전하지 않은 것으로 밝혀졌다.

본 학위 논문에서는 GGH와 CLT 다중선형함수의 대수적 구조를 활용하여 다중선형함수 및 이로부터 파생된 문제를 분석하는 연구를 진행한다.

우선 GGH 다중선형함수에서 낮은 등급의 0의 인코딩이 주어지는 경우, 주어진 아이디얼 격자에서 빠른 시간 안에 짧은 길이의 벡터를 찾는 연구를 진행하여 기존에 알려진 GGH 다중선형함수와 다른 공격 알고리즘을 제안하고, 그 결과 GGH 다중선형함수가 안전하지 않음을 보인다.

두 번째로 낮은 등급의 0의 인코딩이 없는 경우에는, 주어진 GGH의 공개 데이터로부터 낮은 등급의 0의 인코딩을 만드는 방법에 대한 연구를 진행한다. 앞서 진행한 연구결과를 적용하여, 낮은 등급의 0의 인코딩이 없는 경우에도 GGH 함수가 주어진 파라미터에서 안전하지 않음을 보인다.

마지막으로 CLT함수의 경우에는 주어진 공개 정보로부터, 비공개 정보들을 고유값(eigenvalue) 으로 갖는 정방행렬을 설계하는 방법에 대해 연구를 진행한다. 이를 확장하여, CLT 함수의 모든 비공개 정보들을 복구함으로서 CLT 다중선형함수가 구조적으로 안전하지 않음을 증명한다.

주요어휘: GGH 다중선형함수, 최소 격자 문제, NTRU문제, 아이디얼 격자, CLT 다중선형함수, 중국인의 나머지 정리 기반 근사 공약수 문제

학번: 2012-20254